

G112/TS/H/94.✓

**ANALISA PENGUKURAN TEKANAN DARAH
PERIODIK KONTINU, DAN OTOMATIS YANG
DIKONTROL OLEH MIKROKOMPUTER
BERBASIS CPU 8088**



RSE
621.391.6
SUC
a-1
1093

Oleh :

AGUS SUSILO

NRP. 288 220 1025

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1993**

**ANALISA PENGUKURAN TEKANAN DARAH
PERIODIK KONTINU, DAN OTOMATIS YANG
DIKONTROL OLEH MIKROKOMPUTER
BERBASIS CPU 8088**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

Pada

Bidang Studi Teknik Elektronika

Jurusan Teknik Elektro

Fakultas Teknologi Industri

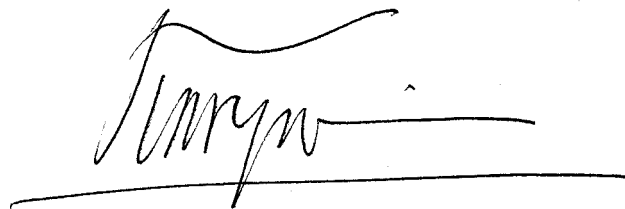
Institut Teknologi Sepuluh Nopember

S u r a b a y a

PERPUSTAKAAN	
ITS	
Tgl. Terima	27 NOV 1993
Terima Dari	A.
No. Agenda Pro.	1654 /TA.

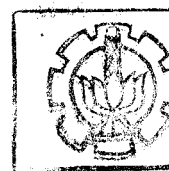
Mengetahui / Menyetujui

Dosen Pembimbing



(Ir. KARYADI, M.Sc.)

**SURABAYA
OKTOBER, 1993**



**MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER**

ABSTRAK

Untuk kasus-kasus tertentu, misalnya dalam proses operasi, maupun pasca operasi, kondisi pasien perlu untuk dimonitor secara kontinu dan otomatis selama periode waktu tertentu.

Tugas akhir ini adalah menganalisa, merancang, dan membuat satu unit alat untuk memantau tekanan darah pasien secara kontinu, dan otomatis, selama periode waktu tertentu.

Dalam pelaksanaan tugas akhir ini kecuali tekanan darah, juga dimonitor heart rate, serta MAP (Mean Artery Pressure).

Monitoring ketiga variabel medis di atas dirancang dengan suatu peralatan yang dikontrol mikrokomputer berbasis 8088.

KATA PENGANTAR

Penulis memanjatkan puja dan puji syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan kasih karunianya, sehingga tugas akhir yang berjudul **ANALISA PENGUKUR TEKANAN DARAH PERIODIK, KONTINU, DAN OTOMATIS YANG DIKONTROL OLEH MIKROKOMPUTER BERBASIS CPU 8088** dapat diselesaikan.

Tugas akhir ini merupakan salah satu syarat untuk mendapatkan gelar sarjana pada Jurusan Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, dengan beban 6 SKS (Satuan Kredit Semester). Penulis menyelesaikan tugas akhir ini dengan berdasarkan pada teori-teori yang didapat dari kuliah dan buku-buku referensi, bimbingan dosen-dosen, pengalaman rekan - rekan, serta dorongan yang diberikan berbagai pihak terutama dorongan dari Papa dan Mama.

Penulis menyampaikan terima kasih yang sebesar-besarnya, kepada:

1. Bapak Ir. Karyadi, Msc, selaku dosen pembimbing penulis yang telah memberikan dorongan dan bimbingan serta mengusahakan buku rujukan yang baik sehingga tugas akhir ini dapat diselesaikan.
2. Bapak Ir. Hendra Kusuma, selaku dosen wali penulis yang memberikan bimbingan dan pengalamannya dalam penyelesaian tugas akhir ini.
3. Bapak Ir. Soetikno, selaku Koordinator Bidang Studi Elektronika, Jurusan Teknik Elektro ITS, yang telah memberikan dorongan dan fasilitas laboratorium mikroelektronika dalam penyelesaian tugas akhir ini.
4. Bapak Ir. Katjuk Astrowulan, MSEE, selaku Ketua Jurusan Teknik Elektro ITS

5. Semua pihak yang telah memungkinkan tugas akhir ini diselesaikan dengan baik.

Akhirnya penulis berharap tugas akhir ini dapat bermanfaat bagi para pembaca, khususnya peminat dan pihak yang berhubungan dengan Elektronika Medika dan Teknik Penerapannya dalam dunia medis.

Surabaya, Oktober 1993

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
I.1. LATAR BELAKANG.....	1
I.2. MASALAH.....	3
I.3. TUJUAN.....	3
I.4. METODOLOGI.....	3
I.5. SISTEMATIKA PENULISAN.....	4
I.6. RELEVANSI.....	4
BAB II TEORI PENUNJANG	
II.1. TEKanan DARAH.....	6
II.1.1. Sistem Vaskuler.....	6
II.1.2. Sistolik Dan Diastolik.....	7
II.1.3. Tekanan Nadi, Tekanan Darah Rata-rata dan Denyut Nadi.....	11
II.2. PENGUKURAN TEKanan DARAH.....	12
II.2.1. Pengukuran Tekanan Darah Secara Langsung.....	12
II.2.1.1. Extravascular Transducer.....	12
II.2.1.2. Intravascular Transducer.....	13
II.2.2. Pengukuran Tekanan Darah Secar Tidak Langsung.....	13
II.2.3. Pengukuran Tekanan Darah Relative.....	15
II.3. TRANSDUCER.....	17
II.3.1. Transducer Bunyi Korotkoff.....	17
II.3.2. Transducer Tekanan.....	19
II.4. BAGIAN PENGOLAH SINYAL.....	19
II.4.1. Amplifier.....	20
II.4.1.1. Inverting Amplifier.....	21
II.4.1.2. Voltage Follower.....	22
II.4.1.3. Non Inverting Amplifier.....	23
II.4.1.4. Differensial Amplifier.....	24
II.4.1.5. Tiga Op - Amp Amplifier.....	25
II.4.2. Filter.....	26
II.4.2.1. Filter Low Pass.....	27
II.4.2.2. Filter High Pass.....	29
II.4.2.3. Filter Band Pass.....	30
II.4.3. Multivibrator Monostable.....	31

II.4.3.1. Multivibrator Monostable Dengan Op - Amp...	31
II.4.3.2. Rangkainya Terpadu Multivibrator Monostable...	33
II.4.4. Multivibrator Astable.....	35
II.4.5. Komparator.....	37
II.5. SISTEM MINIMUM 8088 DAN PERIPHERAL PENDUKUNGNYA.....	39
II.5.1. Mikroprosesor 8088.....	39
II.5.2. Programmable Peripheral Interface 8255.....	44
II.5.3. Programmable Interval Timer.....	48
II.5.4. Universal Synchronous Asynchronous Receiver Transmitter 8251.....	53
II.6. SISTEM PNEUMATIK.....	68
II.6.1. Cuff.....	68
II.6.2. Katup Udara.....	69
II.6.3. Pompa Udara / Kompresor.....	70
II.6.4. Penahanan Aliran Udara.....	70

BAB III PERENCANAAN DAN PEMBUATAN HARDWARE

III.1. PENGUKUR TEKANAN DARAH OTOMATIS< PERIODIK DAN KONTINU.....	72
III.1.1. Bagian Transducer Tekanan.....	72
III.1.1.1. Astable Multivibrator.....	73
III.1.2. Bagian Transducer Bunyi.....	75
III.1.2.1. Amplifier.....	76
III.1.2.2. Filter.....	77
III.1.2.3. Komparator.....	79
III.1.2.4. Monostable Multivibrator.....	79
III.1.3. Sistem Minimum 8088.....	80
III.1.4. Timed Interrupt Generator.....	80
III.1.5. Programmable Peripheral Interface.....	82
III.1.6. Bagian Pneumatik Dan Cuff.....	82
III.1.7. Bagian Input / Output.....	83
III.2. KOMUNIKASI SERIAL DENGAN IBM PC.....	86

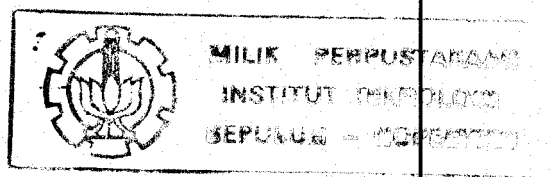
BAB IV PERENCANAAN DAN PEMBUATAN SOFTWARE

IV.1. SOFTWARE MINIMUM SISTEM 8088.....	88
IV.1.1. Prosedur Inisialisasi Hardware.....	88
IV.1.2. Prosedur Masukan Data.....	88
IV.1.3. Prosedur Tampilan Kata.....	90
IV.1.4. Prosedur Scanning Keyboard.....	90
IV.1.5. Prosedur Penundaan Waktu.....	90
IV.1.6. Prosedur Tampilan Alat.....	90
IV.1.7. Prosedur Bunyi speaker.....	91
IV.1.8. Prosedur Komunikasi Serial RS 232.....	91
IV.1.9. Prosedur Pengukuran Tekanan Darah.....	91
IV.1.10. Prosedur Penyiapan Data Untuk RS 232.....	93
IV.1.11. Prosedur Pengambilan Keputusan.....	93
IV.1.12. Prosedur Tampilan Kalimat.....	94
IV.1.13. Prosedur Harga Tekanan Awal Cuff.....	94
IV.1.14. Prosedur Tekanan Cuff.....	94
IV.1.15. Prosedur Tampilan Numerik.....	94
IV.1.16. Prosedur Frekuensi Transducer.....	95
IV.1.17. Prosedur Konversi Frekuensi Ke Tekanan.....	95

IV.1.18. Prosedur NMI.....	96
IV.1.19. Prosedur Buka Katup.....	96
IV.2. SOFTWARE APLIKASI KE KOMPUTER PC.....	96
BAB V PENGUKURAN DAN UJI COBA ALAT	
V.1. BAGIAN TRANSDUCER TEKANAN.....	98
V.1.1. Transducer Tekanan.....	98
V.1.2. Output Frekuensi Astable Multivibrator.....	98
V.2. BAGIAN TRANSDUCER BUNYI.....	102
V.2.1. Transducer Piezoelectrik.....	102
V.2.2. Amplifier.....	102
V.2.3. Filter.....	103
V.3. BAGIAN PNEUMATIS.....	104
V.4. PENGUKURAN TEKANAN.....	105
V.5. UJI COBA ALAT.....	106
BAB VI PENUTUP	
VI.1. KESIMPULAN.....	109
VI.2. SARAN.....	110
DAFTAR PUSTAKA	111
LAMPIRAN A	
DATA HASIL PENGUKURAN	
LAMPIRAN B	
LISTING PROGRAM APLIKASI KOMPUTER PC DAN EPROM	
LAMPIRAN C	
SKEMA RANGKAIAN	
LAMPIRAN D	
USULAN TUGAS AKHIR DAN DAFTAR RIWAYAT HIDUP	

DAFTAR GAMBAR

2.1.	Perubahan Tekanan Aorta Selama Satu Siklus Jantung.....	8
2.2.	Daerah Harga Normal Tekanan Darah Berdasarkan Usia dan jenis kelamin.....	9
2.3.	Contoh Fluktuasi Tekanan Darah Terhadap Waktu.....	10
2.4.	Sistem Extra Vascular Pressure Transducer.....	13
2.5.	Sistem Pengukuran Tekanan Darah Secara Tidak Langsung.....	15
2.6.	Arterial Tonometer Dan Penampang Pergelangan Tangan.....	16
2.7.	Respon Frekuensi Dari Kristal dan Dinamik Mikropon.....	17
2.8.	Piezoelektrik Transducer dan Rangkaian Ekuivalennya.....	18
2.9.	Struktur Dasar Transducer Tekanan.....	19
2.10.	Simbol Elektronika Op-Amp.....	21
2.11.	Inverting amplifier.....	22
2.12.	Voltage Follower Dengan Op-Amp.....	22
2.13.	Non Inverting Amplifier.....	23
2.14.	Defferensial Amplifier.....	24
2.15.	Tiga Op-Amp Amplifier.....	25
2.16.	Filter Low Pass -40 dB / dec.....	28
2.17.	Tanggapan Frekuensi Untuk Filter Low Pass -40 dB / dec.....	28
2.18.	Filter High Pass 40 db / dec.....	29
2.19.	Tanggapan Frekuensi Untuk Filter High Pass 40 dB / dec.....	30
2.20.	Rangkaian Multivibrator Satu Tembakan Dengan Op-Amp.....	31
2.21.	Diagram Waktu Multivibrator Satu Tembakan.....	33
2.22.	Rangkaian Terpadu Multivibrator Satu Tembakan.....	34
2.23.	Multivibrator Astable Dengan Gerbang Nor.....	35
2.24.	Definisi Waktu Tunda Propagasi Dan Transisi.....	35
2.25.	Timing Diagram Multivibrator Astable Nor Gate.....	36
2.26.	Blok Diagram Sistem Minimum.....	39
2.27.	Blok Diagram Fungsi Dan Konfigurasi Pin 8088.....	39
2.28.	Timing Diagram Dari Sistem CPU 8088.....	40
2.29.	Blok Diagram Dari PPI 8255.....	46
2.30.	Blok Diagram PIT 8253.....	49
2.31.	Timing Diagram 8253 Untuk Masing-masing Mode.....	52
2.32.	Blok Diagram Dan Pin-pin USART 8251A.....	55
2.33.	(a) Hubungan Null Modem, (b) Hubungan RS 232C Tanpa Sinyal Handshaking.....	67
2.34.	Interfacing Antara TTL DAN RS-232C	67
2.35.	Efek Dari Penggunaan Cuff.....	69
2.36.	(a) Katup Normally Closed, (b) Katup Normally Open.....	69
2.37.	Pompa / Kompresor Udara.....	70
2.38.	Penahan Aliran Udara.....	71
3.1.	Blok Diagram Sistem Pengukur Tekanan Darah.....	72
3.2.	Bagian Transducer Tekanan.....	73
3.3.	Rangkaian Astable Multivibrator Untuk Transducer Tekanan.....	74
3.4.	Counter Untuk Bagian Transducer Tekanan.....	75
3.5.	Bagian Transducer Bunyi.....	76
3.6.	Rangkaian Amplifier.....	77
3.7.	Filter Low Pass -40 dB / dec.....	78
3.8.	Filter High Pass 40 db / dec.....	78



3.9.	Rangkaian Komparator.....	79
3.10.	Rangkaian Monostable Multivibrator.....	80
3.11.	Rangkaian Penghsail Sinyal Interrupt.....	81
3.12.	Bagian Pneumatik Dan Cuff.....	83
3.13.	Rangkaian Pompa Udara Dan Katup.....	83
3.14.	Rangkaian Tampilan Tiga Digit.....	84
3.15.	Rangakaian Catu daya.....	86
3.16.	Rangkaian Komunikasi Serial RS-232.....	87
4.1.	Flowchart Software Program Eprom.....	89
4.2.	Flowchart Aplikasi Komputer PC.....	97
5.1.	Grafik Tekanan Terhadap Frekuensi.....	101
5.2.	Bentuk Sinyal Output Astable Multivibrator Pada Tekanan 0 mmHg (gambar atas) Dan 300 mmHg (gambar bawah).....	101
5.3.	Output Filter (atas) Dan Monostable (bawah) Saat Pulsa Sistolik Terdeteksi.....	103
5.4.	Output Filter (atas) Dan Monostable (bawah) Saat Pulsa Diastolik Terdeteksi.....	104
5.5.	Tampilan Grafik Hasil Pengukuran.....	106
5.6.	Transducer Bunyi Dan Tekanan.....	107
5.7.	Tampak Depan Dan Belakang Alat.....	107
5.8.	Pengukur Tekanan Darah.....	108

DAFTAR TABEL

2.1.	Harga Rata-rata Sistolik Dan Diastolik Berdasarkan Usia Dan Jenis Kelamin.....	10
2.2.	Sistem Decoding Dari Status Bus Cycle.....	44
2.3.	Format Control Word Untuk PPI 8255.....	47
2.4.	Address Decoding Pada PPI 8255.....	48
2.5.	Address Decoding Pada PIT 8253.....	50
2.6.	Format Coontrol Word PIT 8253.....	50
2.7.	Uraian Format Control Word PIT 8253.....	51
2.8.	Format Bit Transmisi Serial Asinkron.....	53
2.9.	Format Mode Instruction 8251A.....	60
2.10.	Format Command Instruction 8251A.....	61
2.11.	Format Status Word 8251A.....	61
2.12.	DB-25P Connector Dan Nama Pin Serta Sinyal-sinyalnya.....	64
3.1.	Tampilan Alfa Numerik.....	85
5.1.	Data Pengukuran Kapasitansi Transducer Tekanan.....	99
5.2.	Data Pengukuran Frekuensi Output Astable Multivibrator.....	100
5.3.	Hasil Pengukuran Tekanan.....	105
5.4.	Hasil Pengukuran Tekanan Darah Seorang Sukarelawan.....	106

BAB I

PENDAHULUAN

I.1. LATAR BELAKANG

Pengukuran tekanan darah seseorang termasuk salah satu prosedur dasar perawatan oleh dokter. Belakangan ini dengan semakin baiknya taraf hidup masyarakat maka penyakit tertentu seperti tekanan darah tinggi menjadi meningkat penderitanya. Jika hal ini tidak diperhatikan maka dapat berakibat fatal. Seperti pendarahan otak, kelainan fungsi jantung dan sebagainya. Salah satu usaha pencegahannya yaitu dengan memeriksa tekanan darahnya secara rutin, sehingga setiap perubahan dapat segera diketahui dan dapat ditentukan langkah-langkah selanjutnya.

Ada dua cara pengukuran tekanan darah yang dapat dilakukan yaitu pengukuran tekanan darah secara langsung dan pengukuran tekanan darah secara tidak langsung. Pada pengukuran secara langsung maka pembuluh darah dibuka dan pengukuran tekanan darah langsung dilakukan oleh transducer yang dihubungkan pada pembuluh darah tadi. Pada pengukuran secara tidak langsung ada beberapa metoda yang dapat dilakukan, salah satunya yaitu metoda auskultasi. Dalam metoda ini lengan pasien dibalut dengan cuff, udara dipompakan ke dalam cuff sehingga tekanan cuff naik sampai diatas prakiraan tekanan sistolik, kemudian tekanan udara dalam cuff diturunkan pelan-pelan (2 - 3 mmHg/s) dan aliran darah pada bagian lengan yang terbalut cuff dimonitor lewat bunyinya dengan mikropon yang ditempatkan pada lengan bagian bawah. Bunyi

dengan mikropon yang ditempatkan pada lengan bagian bawah. Bunyi Korotkoff yang terdeteksi pertama kali menandakan tekanan cuff saat itu adalah tekanan sistolik. Bunyi korotkoff timbul karena turbulensi aliran darah dan vibrasi dari pembuluh darah pada bagian lengan yang terbalut cuff, bunyi ini timbul secara berkala sesuai dengan periode denyutan jantung, sehingga dengan mengamati jumlah bunyi tiap satuan waktu dapat dihitung denyutan jantung tiap menit (Heart Rate). Karena tekanan cuff diturunkan terus maka suatu saat tekanan dalam pembuluh darah pada bagian lengan yang terbalut cuff selalu lebih tinggi dari tekanan cuff, sehingga aliran darah menjadi laminar, dan tak ada vibrasi pembuluh darah. Pada saat itu bunyi korotkoff melemah dan hilang, tekanan cuff saat itu adalah tekanan diastolik. Tekanan darah rata-rata (Mean Artery Pressure) dihitung dari sistolik dan diastolik.

Pada prakteknya sering diperlukan harga tekanan darah secara periodik dalam kontinuitas waktu tertentu dan pengukurannya berlangsung secara otomatis. Untuk memenuhi keperluan tersebut maka dapat digunakan sistem minimum 8088, dengan sistem minimum data hasil pengukuran juga dapat dikirim ke komputer untuk pengolahan lebih lanjut. Dewasa ini data-data medis sering juga dimasukkan ke sistem mikrokomputer untuk mendiagnosis keadaan pasien berdasarkan data-data hasil pengukuran dan data-data medis tersebut sehingga keadaan pasien dapat dilihat berdasarkan perbandingan tersebut. Pada tugas akhir ini dicoba menentukan keadaan tekanan darah pasien berdasarkan data-data medis tekanan darah normal menurut usia dan jenis kelamin.

I.2. MASALAH

Masalah yang akan diselesaikan yaitu melakukan otomatisasi proses pengukuran tekanan darah dengan metoda auskultasi, pengaturan sistem yang dirancang agar dapat bekerja secara periodik dengan kontinuitas waktu tertentu, menampilkan hasil pengukuran dan diagnosa, serta mengirim data-data hasil pengukuran ke komputer untuk pengolahan lebih lanjut.

I.3. TUJUAN

Tujuannya yaitu menganalisa sistem pengukuran tekanan darah yang otomatis, periodik, dan kontinu. Analisa ini meliputi cara mendapatkan harga tekanan udara dalam cuff, cara mendeteksi bunyi korotkoff, proses untuk pengukuran tekanan darah, pengendalian sistem sehingga dapat bekerja periodik dalam jangka waktu tertentu, membandingkan data-data hasil pengukuran dengan data-data medis (dalam hal ini range tekanan darah normal berdasarkan usia dan jenis kelamin) untuk menentukan hasil diagnosa, proses pengiriman data-data hasil pengukuran ke komputer dan pengolahan lebih lanjut di komputer. Pada akhirnya merealisasi sistem dari hasil analisa tersebut.

I.4. METODOLOGI

Metodologi yang dijalankan untuk mencapai tujuan seperti di atas adalah:

1. Melakukan studi literatur tentang alat pengukuran tekanan darah secara tak langsung dengan metoda auskultasi.
2. Membuat perencanaan bagian-bagian yang menyusun alat pengukur

tekanan darah sesuai dengan hasil analisa sistem yang akan dibentuk.

3. Merealisasikan bagian-bagian tersebut dan menggabungkan sehingga terbentuk alat / sistem.
4. Melakukan uji coba.
5. Menyusun buku Tugas Akhir.

I.5. SISTEMATIKA PENULISAN

Buku tugas akhir ini disusun dengan sistematika sebagai berikut: Bab I merupakan Pendahuluan yang berisi latar belakang, permasalahan, tujuan, metodologi, sistematika dan relevansi dari alat yang akan dibuat. Bab II berisi teori penunjang tentang elektronika medika dan instrumentasi elektroniknya, serta analisa alat pengukur tekanan darah secara tak langsung dengan metoda auskultasi. Bab III menguraikan perencanaan dan pembuatan hardware. Bab IV menguraikan perencanaan dan pembuatan software. Bab V menguraikan pembuatan dan uji coba alat, Bab VI merupakan kesimpulan dan penutup dari buku Tugas Akhir ini.

I.6. RELEVANSI

Tugas akhir ini bermanfaat untuk memeriksa keadaan kesehatan seseorang lewat tekanan darahnya, karena alat bersifat periodik dan kontinu maka keadaan tekanan darah dapat dimonitor terus selama periode tertentu, hal ini sering diterapkan dalam operasi maupun untuk pasien-pasien di ruang gawat darurat dan ruang recovery. Alat yang dibuat juga dapat menampilkan hasil diagnosa yang akan dapat dipakai

rujukan oleh tenaga medis maupun orang awan yang tidak mengerti jangkauan normal tekanan darah berdasar umur dan jenis kelamin.

Dalam proses pembuatan tugas akhir ini dapat dilihat langkah-langkah merealisasi suatu alat elektronik, dimulai dari analisa alat dan bagian-bagiannya untuk tujuan / fungsi tertentu, merencanakannya, merealisasikan, melakukan uji coba dan mendapatkan ide untuk penyempurnaan lebih lanjut.

BAB II

TEORI PENUNJANG

II.1. TEKANAN DARAH

II.1.1. Sistem Vaskuler

Darah mempunyai peranan yang sangat penting bagi tubuh, mulai dari membawa zat-zat yang diperlukan untuk proses metabolisme seperti zat-zat makanan dan oksigen, membawa zat-zat yang berfungsi sebagai pengatur dan pemelihara tubuh seperti hormon-hormon dan zat-zat yang berguna untuk mempertahankan tubuh dari serangan penyakit, maupun membawa zat-zat sisa dan hasil dari proses metabolisme yang harus segera dibuang. Seluruh bagian tubuh yang hidup memerlukan darah, oleh karena itu darah harus disirkulasikan keseluruh bagian tubuh lewat sistem vaskuler. Sistem vaskuler dapat dibagi dalam tiga bagian, yaitu¹:

1. Sistem distribusi yang terdiri dari arteri dan arteriola dengan fungsinya sebagai penyalur darah ke seluruh organ dan jaringan tubuh, sistem ini mengatur alirannya keseluruh bagian tubuh yang memerlukan darah.
2. Sistem difusi yang terdiri dari pembuluh darah kapiler, yang ditandai dengan dindingnya yang tersusun sedemikian

¹Ibnu Masud, MS, Dr, Dasar-dasar Fisiologi Kardiovaskular, EGC, Jakarta, 1989, hal 102

rupa sehingga memungkinkan proses difusi suatu bahan berlangsung didalamnya seperti: karbon dioksida, oksigen, zat gizi dan sisa-sisa metabolisme serta tidak jarang sel-sel darah juga dapat melewatinya. Sistem ini bersama-sama dengan arteriola berfungsi memelihara tahanan vaskuler.

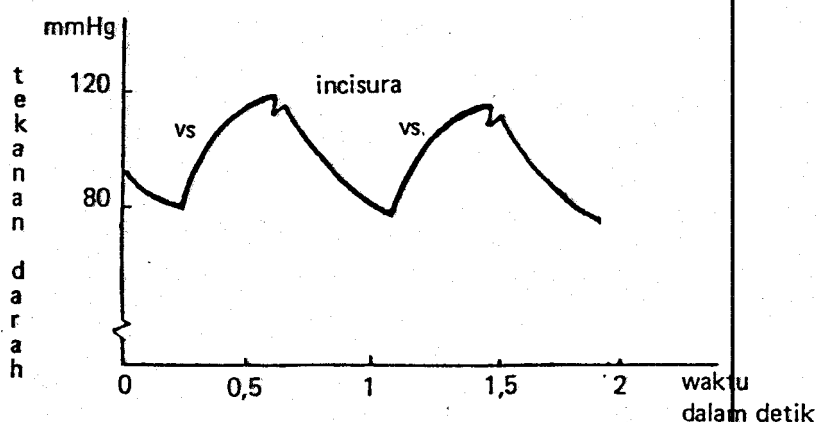
3. Sistem pengumpul, yang berfungsi menerima dan mengumpulkan darah dari kapiler, pembuluh limfe, atau langsung dari sistem arteri. Bagian ini mengalirkan darah kembali ke jantung. (sistem ini merupakan pembuluh darah balik / vena)

Susunan fisiologi-anatomi sistem vaskuler tersebut merupakan sistem saluran tertutup, dengan darah yang terdapat dalam pembuluh vena dapat dipompakan oleh jantung ke sistem pembuluh darah arteri dan pada akhirnya kembali lagi ke sistem vena. Darah mengalir di dalam sistem vaskuler karena tingkah laku jantung yang berkontraksi dan berelaksasi, sehingga menimbulkan perubahan-perubahan tekanan yang mampu memompa darah dari dan akhirnya kembali ke jantung.

II.1.2. Sistolik Dan Diastolik

Aktivitas jantung yang memompa darah dengan cara mengadakan kontraksi dan relaksasi, menimbulkan perubahan tekanan darah di dalam sistem sirkulasinya. Pada waktu sistole ventrikel (yaitu fase ejeksi cepat)² darah dipompa ke aorta dan arteri paru. Tekanan arteri pada

saat itu naik sampai sekitar 120 mmHg, tekanan ini yang dalam pengukuran tekanan darah disebut sebagai **tekanan sistolik**, kenaikan tekanan ini menyebabkan aorta mengalami distensi, sehingga tekanan di dalamnya turun sedikit. Jika dilihat pada grafik tekanan terhadap waktu maka akan terlihat sebagai incisura atau celah akibat penurunan tekanan sesaat dan disusul dengan kenaikannya kembali. Pada saat diastole ventrikel, maka tekanan aorta cenderung turun sampai dengan sekitar 80 mmHg, tekanan inilah yang pada pemeriksaan tekanan darah dikenal sebagai **tekanan diastolik**. Perubahan tekanan darah terhadap waktu ditunjukkan pada gambar 2.1.

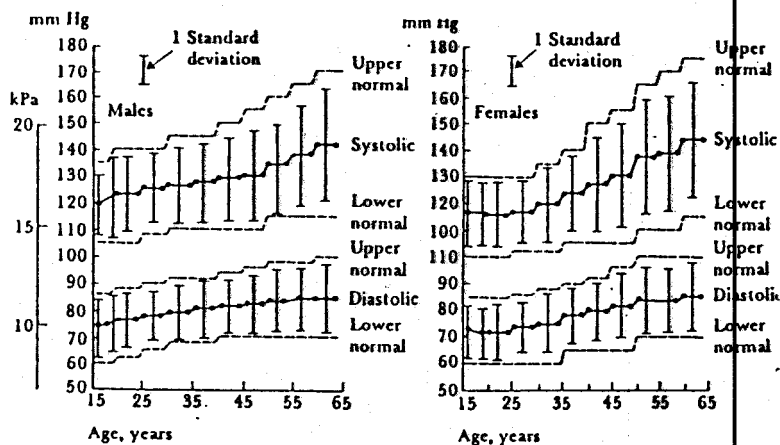


Gambar 2.1. Perubahan Tekanan Aorta Selama Satu Siklus Jantung

Tekanan normal pada orang dewasa sangat bervariasi. Tekanan darah terdiri dari tekanan sistolik yang berkisar dari 95 mmHg sampai 140 mmHg, tekanan ini dapat meningkat dengan bertambahnya usia. Di lain pihak tekanan diastolik berkisar dari 60 mmHg sampai 90 mmHg. Namun tekanan darah normal pada umumnya berkisar 120 mmHg untuk

tekanan sistolik dan 80 mmHg untuk tekanan diastolik³.

Tekanan darah normal dipengaruhi oleh beberapa faktor antara lain oleh usia, jenis kelamin, perjalanan waktu dalam sehari semalam yang membentuk periode perubahan tekanan darah. Tekanan darah normal untuk usia dan jenis kelamin tertentu bukan merupakan harga yang tertentu, tapi berupa suatu range tertentu, seperti diperlihatkan pada gambar 2.2.



Gambar 2.2. Daerah Harga Normal Tekanan Darah Berdasarkan Usia dan Jenis kelamin⁴

Range tekanan darah normal menurut usia dan jenis kelamin berdasarkan rekomendasi dari WHO ditabelkan sbb:

(tabel dikutip dari Instruction Manual Omron Automatic Digital Blood

³ ibid, hal. 111

⁴ John G. Webster, Medical Instrumentation Application And Design, Houghton Mifflin Company, boston, 1978, hal. 375

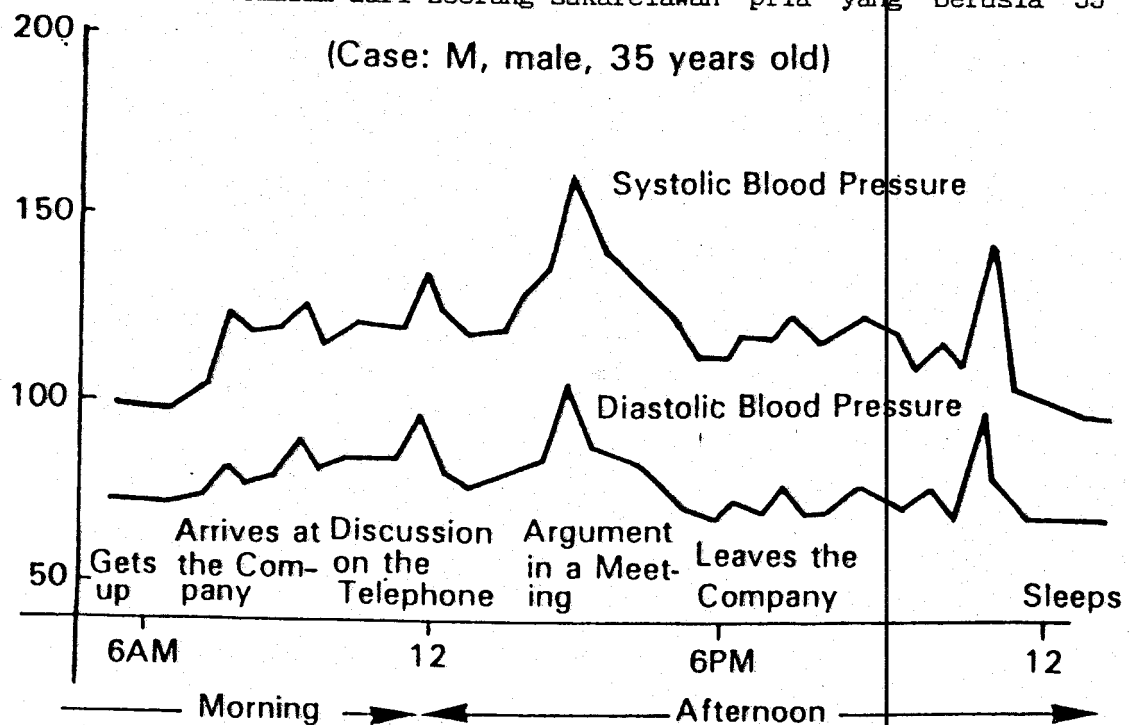
Pressure Monitor)

Tabel 2.1. Harga Rata-rata Sistolik Dan Diastolik Berdasarkan Usia Dan Jenis Kelamin

AGE	MALE		FEMALE	
	SYSTOLIC	DIASTOLIC	SYSTOLIC	DIASTOLIC
20-24	128 ± 14	75 ± 13	121 ± 13	72 ± 12
25-29	128 ± 14	75 ± 12	122 ± 15	73 ± 12
30-34	129 ± 16	77 ± 14	124 ± 15	75 ± 12
35-39	130 ± 18	79 ± 15	127 ± 17	78 ± 13
40-44	132 ± 19	81 ± 14	132 ± 20	80 ± 13
45-49	136 ± 22	83 ± 14	140 ± 26	84 ± 16
50-54	144 ± 26	87 ± 16	147 ± 28	86 ± 15
55-59	150 ± 27	88 ± 16	150 ± 28	88 ± 16
60-64	156 ± 28	91 ± 16	158 ± 30	90 ± 16
65-69	158 ± 30	89 ± 17	166 ± 30	91 ± 15
OVER 70	165 ± 32	89 ± 17	171 ± 31	91 ± 16

Unit: mmHg

Gambar 2.3. Memperlihatkan fluktuasi tekanan darah terhadap waktu dalam sehari semalam dari seorang sukarelawan pria yang berusia 35



Gambar 2.3. Contoh Fluktuasi Tekanan Darah terhadap Waktu

tahun (grafik dikutip dari Instruction Manual Omron Automatic Digital Blood Pressure Monitor).

II.1.3. Tekanan Nadi, Tekanan Darah Rata-rata, Dan Denyut Nadi

Selisih antara tekanan sistolik dan tekanan diastolik dikenal dengan tekanan nadi, jadi jika tekanan sistolik sebesar 120 mmHg dan tekanan diastolik sebesar 80 mmHg, maka tekanan nadinya sama dengan 40 mmHg. Karena tekanan darah pada umumnya tidak tetap karena terpengaruh beberapa faktor, maka tekanan nadinya juga berubah - ubah selaras dengan tekanan darahnya.

Tekanan darah rata-rata atau sering disebut Mean Artery Pressure (MAP) adalah tekanan rata-rata diseluruh sistem arteri pada satu siklus jantung. Tekanan darah rata-rata diperoleh dengan cara membagi tekanan nadi dengan tiga dan ditambahkan pada tekanan diastolik,⁵ sesuai dengan rumus:

$$MAP = 1/3 * (Ts - Td) + Td$$

dimana Ts adalah tekanan sistolik dan Td adalah tekanan diastolik.

Selama pembedahan ahli anestesi secara kontinu mengamati MAP untuk mendeteksi kegagalan sirkulasi. Jika MAP menjadi terlalu rendah, hal ini menunjukkan terlalu banyak obat anestesi.

Denyut nadi merupakan manifestasi penalaran perubahan tekanan pada waktu sistole ventrikel, denyut nadi dapat dirasakan dengan perabaan ujung-ujung jari pada daerah dimana arteri letaknya dekat

⁵ Ibnu Masud, op.cit, hal. 112

dibawah kulit, seperti pada pergelangan tangan bagian radial arteri radialis, di daerah leher terdapat arteria carotis Externa, dsb. Jumlah denyut jantung yang juga dapat dirasakan sebagai denyut nadi tiap waktu (menit) dinamakan Heart Rate.

II.2. PENGUKURAN TEKANAN DARAH

II.2.1. Pengukuran Tekanan Darah Secara Langsung

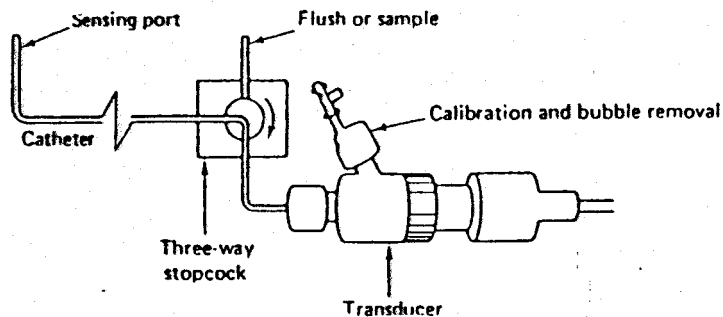
Pengukuran tekanan darah secara langsung dilakukan dengan mengamati tekanan dalam pembuluh darah. Ada dua macam jenis transducer yang digunakan yaitu extravascular pressure transducer dan intravascular pressure transducer.⁶ Sedangkan jenis elemen transducer yang digunakan, diantaranya: strain gage, linear variable differential transformer, variable inductance, variable capacitance, electro-optical, piezoelectric, dan semiconductor device.

II.2.1.1. Extravascular Transducer

Sistem transducer extravascular diperlihatkan pada gambar 2.4. Sistem ini terdiri atas kateter yang dihubungkan ke 'three-way stopcock', lalu ke rumah dari transducer tekanan. Kateter dari sistem transducer diisi dengan larutan saline-heparine, yang harus disemur dengan cairan / larutan setiap beberapa menit supaya darah tak menggumpal pada bagian ujung dari kateter yang dimasukkan ke pembuluh darah. Tekanan darah dalam pembuluh darah diteruskan lewat cairan

⁶ John G. Webster, opcit, hal 336

dalam kateter, lalu ke rumah transducer dan akhirnya ke elemen transducernya sendiri.

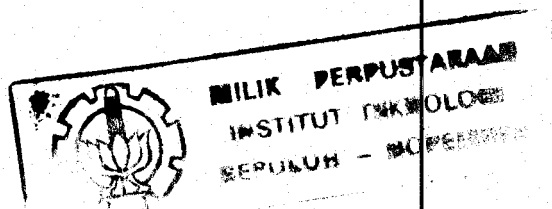


Gambar 2.4. Sistem Extravascular Pressure Transducer

II.2.1.2. Intravascular Transducer

Transducer yang ditempatkan pada ujung kateter mempunyai keuntungan dihilangkannya perantara hidrolis antara sumber tekanan dalam ini tekanan dalam pembuluh darah dan elemen transducer. Pada sistem extravascular transducer, respon frekuensi dari sistem dibatasi oleh kemampuan cairan hidrolis yang berfungsi sebagai penghubung tadi. Mendeteksi tekanan pada ujung kateter tanpa lewat cairan perantara membuat pengukuran mempunyai respon frekuensi yang tinggi dan menghilangkan delay dari sumber tekanan ke elemen transducer. Banyak jenis dari transducer yang digunakan untuk mendeteksi tekanan pada ujung kateter, diantaranya jenis strain gage yang dilekatkan pada selaput yang fleksible pada ujung kateter. Kerugian pemakaian sistem intravascular transducer adalah harganya yang mahal dan kadang transducer yang terpasang di ujung kateter cepat rusak.

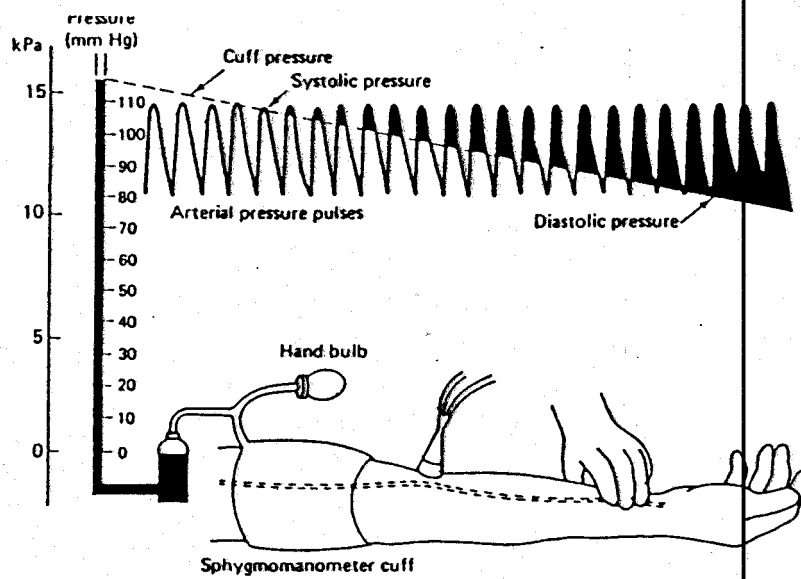
II.2.2. Pengukuran Tekanan Darah Secara Tidak Langsung



Pengukuran tekanan darah secara tak langsung dilakukan dengan *sphygmomanometer*. Gambar 2.5. memperlihatkan metoda pengukuran jenis ini. Sphygmomanometer cuff diberi tekanan udara sehingga aliran darah dalam arteri yang pada anggota tubuh (biasanya lengan) yang dilingkupi oleh cuff tadi menjadi terputus atau terhalang. Tekanan dalam cuff pelah-pelan diturunkan dan pada saat tekanan berada di bawah tekanan sistolik maka darah mulai menyembur melewati bagian arteri yang tertekan oleh cuff tadi, aliran yang bersifat turbulen ini menimbulkan suara korotkoff, yang dapat didengar lewat stetoskop. Ketika tekanan cuff jatuh di bawah tekanan diastolik maka tak ada lagi bunyi korotkoff yang terdengar karena pada saat itu sudah tak ada semburan darah dan aliran yang ada sudah bersifat laminar. Metoda determinasi tekanan darah seperti cara diatas disebut metoda auskultasi dan metoda ini cocok untuk diotomatisasi. Prosedur yang dilakukan adalah memompa cuff secara cepat sehingga tekanannya berada 30 mmHg diatas tekanan sistolik yang diperkirakan. Cuff kemudian di dikurangi tekanannya secara pelan-pelan (2 - 3 mmHg / S)⁷ sampai bunyi Korotkoff pertama terdeteksi oleh mikropon yang diletakkan di bawah cuff. Kemudian harga tekanan cuff pada saat itu disimpan di memori. Cuff terus diturunkan tekanannya sampai suara korotkoff melemah dan hilang terdeteksi dan tekanan cuff pada saat itu disimpan pula pada memori. Harga tekanan cuff tadi ditampilkan sebagai harga

⁷ John G. Webster, opcit, hal 373

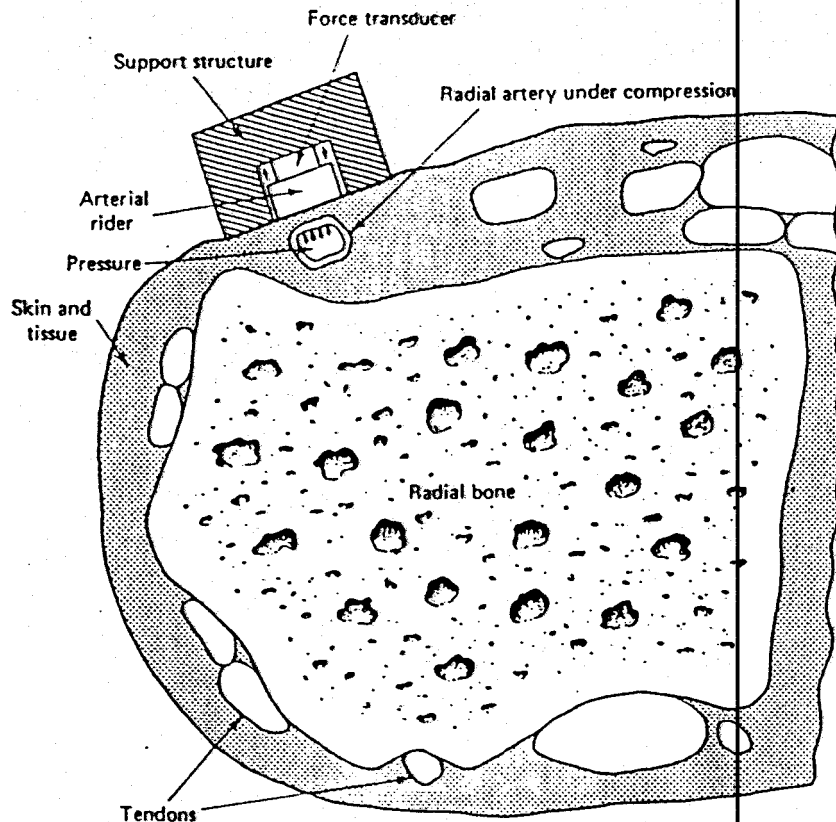
sistolik dan diastolik. Beberapa metoda lain juga dikembangkan untuk mengganti metoda bunyi korotkoff ini diantaranya yaitu metoda yang mengamati pulsa tekanan di dalam cuff, yang lainnya menggunakan ultrasonic untuk mendeteksi gerakan dinding arteri, dsb.



Gambar 2.5. Sistem Pengukuran Tekanan Darah Secara Tidak Langsung

II.2.3. Pengukuran Tekanan Darah Relative

Ketika suatu obyek yang ada diluar pembuluh darah mengurangi tekanannya pada pembuluh yang fleksibel tersebut maka tekanan dalam pembuluh darah tersebut akan mengimbangnya sehingga tekanan dari obyek (tekanan eksternal) akan sama dengan tekanan dalam pembuluh darah tadi, ini adalah prinsip dasar dari Tonometry. Gambar 2.6. memperlihatkan tonometer yang digunakan menekan radial arteri yang ada di pergelangan. Tekanan arteri adalah sebanding dengan gaya tekan pada bidang tekan dibagi luas bidang tekan tadi. Tonometer yang didisain



Gambar 2.6. Arterial Tonometer Dan Penampang Pergelangan Tangan

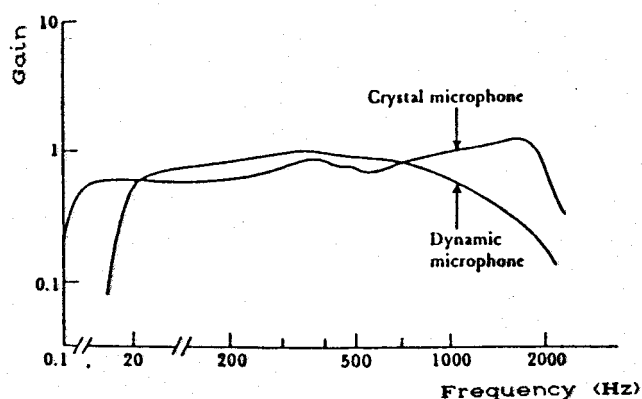
sekarang hanya mampu mengukur tekanan relative saja, karena itu masih memerlukan kalibrasi dengan pengukuran langsung maupun tak langsung. Hal ini terjadi karena bidang tekan dari transducer sebelum menekan arteri juga menekan jaringan tubuh diantaranya sehingga tekanan transducer sebenarnya lebih besar dari tekanan arteri. Oleh karena itu harus dilakukan kalibrasi pada skala yang akan digunakan tonometer tersebut. Tonometer sering tidak bekerja dengan baik karena adanya sedikit tendon ataupun jaringan yang berlebihan yang ada diantara

transducer dan arteri yang akan diukur tekanannya⁸.

II.3. TRANSDUCER

II.3.1. Transducer Bunyi Korotkoff

Ada beberapa transducer yang peka terhadap bunyi, diantaranya piezoelectric (kristal mikropon) dan dinamik mikropon. Gambar 2.7. memperlihatkan respon frekuensi untuk kedua transducer tersebut.



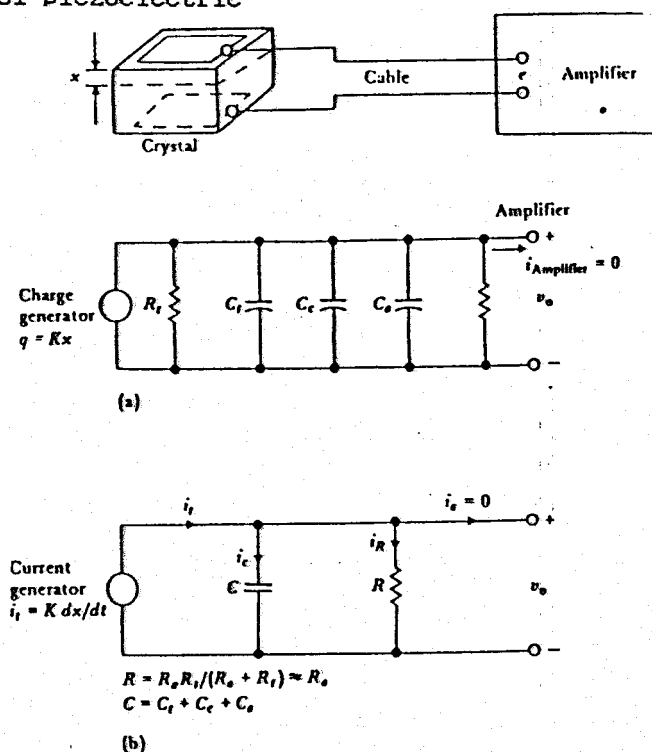
Gambar 2.7. Respon Frekuensi Dari Kristal Dan Dinamik Mikropon

Dari gambar terlihat kristal mikropon lebih sensitive pada frekuensi rendah, tetapi dinamik mikropon mempunyai tegangan output lebih tinggi. Karena untuk mendeteksi bunyi korotkoff dibutuhkan respon yang baik pada frekuensi rendah, maka transducer bunyi yang akan digunakan yaitu kristal mikropon.

Kristal mikropon termasuk jenis piezoelectric transducer. Bahan

⁸ Willis J. Tompkins and John G. Webster, Design Of Microcomputer-Based Medical Instrumentation, Prentice Hall, Inc, Englewood Cliffs, New Jersey, 1981, hal. 49

piezoelectric menghasilkan potensial listrik jika terjadi perubahan yang disebabkan oleh tarikan atau tegangan mekanik pada bahan, dan juga sebaliknya terjadi jika bahan diberi potensial listrik maka akan mengalami perubahan mekanik (tarikan, tegangan mekanik). Gambar 2.8. memperlihatkan bahan piezoelectric dan rangkaian ekivalennya. Tegangan output dari pizeoelectric seluas 1 cm persegi dengan ketebalan 1 mm dan dikenakan gaya berat 10 g adalah sekitar 0.23 mV untuk bahan kristal, sekitar 14 mV untuk bahan barium titanate⁹. Impedansi output sangat tinggi (bisa mencapai 100 GΩ) dan frekuensi low-corner f_c , dapat dinyatakan: $f_c = 1/2\pi RC$, dimana R impedansi input amplifier, C kapasitansi piezoelectric¹⁰



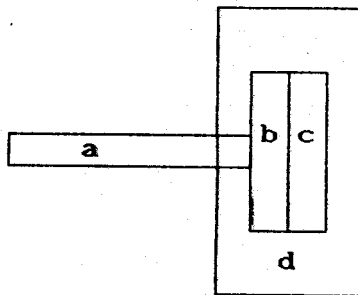
Gambar 2.8. Piezoelectric Transducer Dan Rangkaian Ekivalennya

⁹John G. Webster, opcit, hal 67

¹⁰John G. Webster, opcit, hal. 69

II.3.2. Transducer Tekanan

Yang dimaksud dengan tekanan pada transducer tekanan yang akan digunakan pada tugas akhir ini adalah tekanan udara. Secara umum struktur dari transducer ini diperlihatkan dalam gambar 2.9.



- a = saluran penghubung ke ruang yang akan diukur tekanannya
- b = ruang yang mempunyai pelat elastis sepanjang perubahan tekanan dalam daerah yang diizinkan
- c = elemen transducer yang digunakan, melekat di pelat
- d = rumah dari transducer tekanan udara, sebagai bagian tempat melekat bagian lainnya yang bersifat harus tetap

Gambar 2.9. Struktur Dasar Transducer Tekanan

Elemen transducer dapat berupa strain gage, induktansi, LVDT, kapasitansi, dsb yang harga besaran listriknya berubah sesuai dengan perubahan tekanan yang masuk pada bagian a.

II.4. BAGIAN PENGOLAH SINYAL

Sebelum dimasukkan ke sistem microcomputer maka besaran - besaran listrik yang diperoleh dari transducer harus diolah sedemikian rupa sehingga besaran tadi berubah menjadi suatu sinyal yang cukup memenuhi syarat untuk diolah lebih lanjut dalam sistem mikrokomputer.

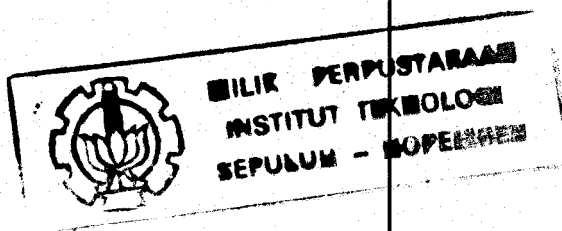
II.4.1. Amplifier

Sinyal listrik yang keluar dari transducer masih lemah (orde μV - mV) maka untuk pemrosesan lanjut sinyal tadi harus diperkuat, untuk itu digunakan rangkaian amplifier.

Ada beberapa parameter penting dari amplifier, diantaranya:

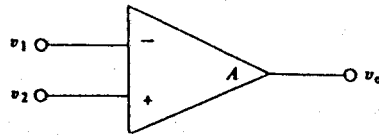
1. Common Mode Gain (CMG), yaitu penguatan amplifier terhadap sinyal Common Mode (sinyal Common Mode yaitu sinyal yang potensialnya semua tempat sama, biasanya terjadi karena medan elektro statis).
2. Differential Gain (DG), yaitu penguatan amplifier terhadap sinyal yang besar potensialnya berbeda yang masuk pada kedua input ampliifiernya.
3. Common Mode Rejection Ratio (CMMR), yaitu besar DG dibagi dengan besarnya CMG, untuk amplifier dicari yang CMMR-nya besar sebab sinyal common mode akan diredam, sinyal Common Mode biasanya tak dikehendaki, contohnya sinyal noise yang terjadi pada kawat panjang yang terinduksi.

Rangkaian amplifier mudah diimplementasikan dengan menggunakan Operational Amplifier (Op - Amp). Op - Amp adalah komponen amplifier yang mempunyai penguatan open loop tinggi (sekitar 200.000 kali), suatu rangkaian amplifier dapat diwujudkan dengan sebuah Op - Amp dan beberapa komponen pasif sebagai penunjangnya. Sebuah Op - Amp mempunyai dua terminal input yaitu masukan tak membalik (+) dan masukan membalik (-), besar outputnya adalah perbedaan tegangan input (+) dan tegangan input (-) dikalikan penguatan Op - Amp tersebut, tegangan output maksimum (jenuh) adalah sebesar $\pm 90\%$ dari tegangan catu dayanya. Simbol elektronika Op - Amp dapat di lihat digambar



BILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPELHIN

berikut.



Gambar 2.10 Simbol Elektronika Op - Amp

Sebuah Op - Amp yang ideal mempunyai karakteristik sbb:

1. Open Loop Gain sama dengan tak terhingga.
2. Tegangan output (V_o) harus sama dengan nol jika $v_2 - v_1 = 0$.
3. Impedansi input adalah tak terhingga.
4. Impedansi output adalah nol.
5. Bandwidth tak terhingga (tak ada batas respon frekuensi dan tak ada pegeseran phase).

Ada dua aturan dasar untuk Op - Amp, yaitu:

1. Bila output Op - Amp masih dalam 'linear range' nya maka tegangan input pada kedua terminal inputnya adalah sama.
2. Tak ada arus yang mengalir ke atau dari kedua terminal input Op - Amp.

II.4.1.1 Inverting Amplifier

Inverting Amplifier adalah suatu amplifier yang mempunyai penguatan negatif, artinya polaritas sinyal input berlawanan dengan polaritas sinyal output. Rangkaian dasarnya dapat dilihat pada gambar 2.11.

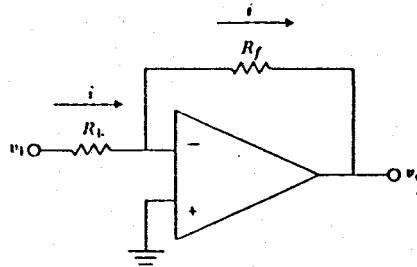
Penurunan Rumus:

$$i_{R_i} = i_{R_f}$$

$$V_i / R_i = (0 - V_o) / R_f$$

$$V_o / V_i = - R_f / R_i$$

$$\text{Close loop gain} = - R_f / R_i$$

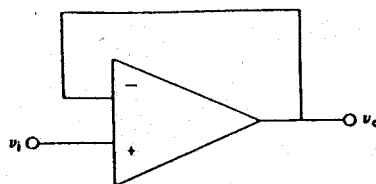


Gambar 2.11 Inverting Amplifier

Kelemahan rangkaian ini yaitu impedansi inputnya rendah ($= R_i$), sehingga kurang cocok dipakai sebagai amplifier sinyal dari sumber yang mempunyai impedansi output tinggi seperti kebanyakan sinyal biopotensial.

II.4.1.2. Voltage Follower

Karena sifat Op - Amp yaitu impedansi inputnya sangat besar maka dapat dibentuk suatu buffer yaitu suatu rangkaian yang mempunyai penguatan satu, namun mempunyai impedansi masukan yang besar dan impedansi keluaran yang rendah dengan hanya menggunakan sebuah Op - Amp.

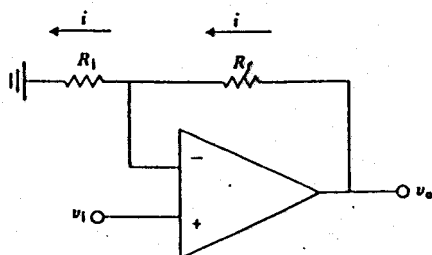


Gambar 2.12 Voltage Follower dengan Op - Amp

Pada rangkaian ini input Op - Amp membalik (-) dihubungkan dengan output Op - Amp, tegangan input dihubungkan dengan masukan Op - Amp tak membalik (+), sesuai dengan sifat Op - Amp maka tegangan pada masukan membalik (-) sama dengan tegangan pada masukan tak membalik (+) sehingga besar tegangan output sama dengan besar tegangan input.

II.4.1.3. Non Inverting Amplifier

Non Inverting Amplifier adalah suatu amplifier yang tak membalikkan polaritas sinyal hasil penguatan. Rangkaianannya seperti gambar 2.13.



Gambar 2.13 Non Inverting Amplifier

Penurunan Rumus:

$$i_{R_i} = i_{R_f}$$

$$(0 - v_i) / R_i = (v_i - v_o) / R_f$$

$$- R_f \cdot v_i = R_i \cdot v_i - R_i \cdot v_o$$

$$R_i \cdot v_o = (R_i + R_f) \cdot v_i$$

$$v_o / v_i = (R_i + R_f) / R_i$$

$$v_o / v_i = 1 + R_f / R_i$$

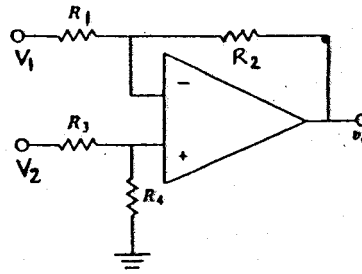
$$\text{Close loop gain} = 1 + R_f / R_i$$

Impedansi input rangkaian Non Inverting Amplifier adalah sebesar

impedansi input Op - Amp nya.

II.4.1.4. Differensial Amplifier

Differensial Amplifier adalah merupakan gabungan dari Non-Inverting dan Inverting Amplifier. Rangkaianya seperti gambar 2.14.



Gambar 2.14 Differensial Amplifier

Penurunan rumus:

Bila diterapkan prinsip superposisi maka berlaku:

$$V_o = V_{o1} + V_{o2},$$

Untuk Inverting Amplifier,

$$V_{o1} = - R_2 / R_1 * V_1$$

Untuk Non Inverting Amplifier

$$\begin{aligned} V_{o2} &= (R_2 + R_1) / R_1 * R_4 / (R_3 + R_4) * V_2 \\ &= (R_2 + R_1) / (R_3 + R_4) * R_4 / R_1 * V_2 \end{aligned}$$

$$\text{Jadi } V_o = - R_2 / R_1 * V_1 + [(R_2 + R_1)/(R_3 + R_4) * R_4/R_1 * V_2]$$

Jika $R_1 = R_3$ dan $R_2 = R_4$, maka:

$$\begin{aligned} V_o &= - R_2 / R_1 * V_1 + [R_2 / R_1 * V_2] \\ &= (V_2 - V_1) * R_2 / R_1, \text{ jika } V_2 - V_1 = V_d, \text{ maka:} \end{aligned}$$

$$V_o = R_2 / R_1 * V_d$$

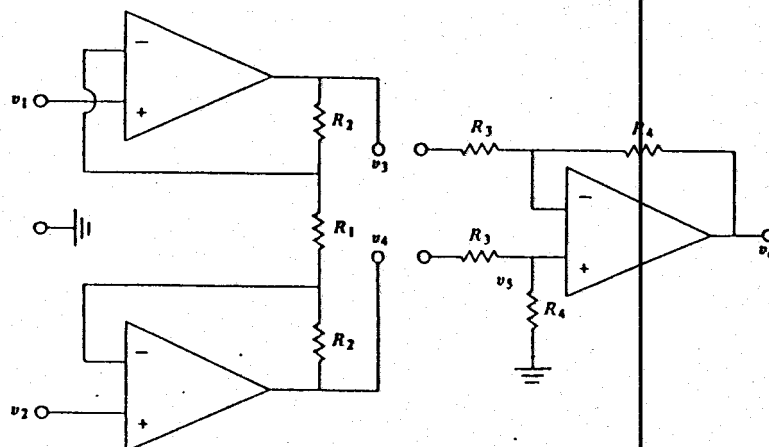
$$\text{Differensial Gain} = R_2 / R_1 * V_d$$

Karena Differensial Amplifier memanfaatkan kedua input Op - Amp sebagai masukan sinyal yang akan diperkuat maka untuk sinyal common mode pada kedua input Op - Amp akan saling menghilangkan sehingga Common Mode Gain nya adalah nol, sedangkan Differensial Gainnya $= R_2 / R_1 * V_d$, karena itu secara teoritis CMRR untuk differensial Amplifier = ∞ .

Oleh karena itu amplifier jenis ini sangat baik untuk sinyal input kecil yang penuh dengan noise sinyal common mode.

II.4.1.5. Tiga Op - Amp Amplifier

Differensial Amplifier mempunyai satu kelemahan yaitu impedansi inputnya tidak terlalu tinggi, sehingga untuk sinyal biopotensial yang sumbernya mempunyai impedansi output relatif tinggi, differensial amplifier harus dimodifikasi sehingga kedua inputnya mempunyai impedansi yang tinggi, untuk itu digunakan voltage follower pada masing - masing inputnya. Rangkaian yang dimaksud dinamakan tiga Op - Amp Amplifier.



Gambar 2.15 Tiga Op - Amp Amplifier

Penurunan rumus:

Untuk bagian input yaitu voltage follower dengan gain, groundnya diambil dari output kedua voltage follower setelah melalui R2,

Differensial Gain (DG) untuk bagian ini:

$$DG = (V3 - V4) / (V1 - V2)$$

$$= (2.R2 + R1) / R1$$

Untuk sinyal common mode pada bagian ini berlaku:

$$V1 = V3, \text{ dan } V2 = V4$$

$$\text{Jadi Common Mode Gain} = 1$$

Differensial Gain untuk Tiga Op - Amp Amplifier:

$$DG_{\text{total}} = Vo / (V1 - V2)$$

$$= (2.R2 + R1) / R1 * - (R4 / R3)$$

$$CMG_{\text{total}} = 1 * 0 \text{ (nol adalah CMG dari bagian differensial Amplifier)}$$

$$\text{Jadi secara teoritis } CMRR_{\text{total}} = \infty$$

II.4.2. Filter

Sinyal yang diinginkan mempunyai frekuensi tertentu, pada prakteknya sinyal yang berasal dari sumbernya (transducer) telah mengalami gangguan pada frekuensi tertentu sesuai dengan sumber noisenya, misalnya gangguan interferensi 50 Hz dari jala - jala listrik, gangguan frekuensi tinggi yang berasal dari pemancar radio, dsb. Ada kalanya sinyal informasi yang diinginkan jauh lebih kecil dibandingkan dengan sinyal yang merupakan noisenya, hal ini tak diinginkan untuk pengolahan sinyal selanjutnya, jika frekuensi sinyal informasi dan frekuensi - frekuensi noisenya berbeda maka untuk mengatasi masalah di atas dapat digunakan filter, filter merupakan

sebuah rangkaian yang dirancang agar melewatkan suatu pita frekuensi tertentu seraya memperlemah semua isyarat di luar pita ini¹¹. Rangkaian filter dapat direalisasikan dengan komponen R, L, C, namun dengan adanya Op - Amp dapat disusun filter aktif yang ukurannya relatif kecil dan mempunyai respon frekuensi (db / dec) yang tajam. Berdasarkan karakteristik frekuensinya maka filter dapat digolongkan sbb:

1. Low Pass Filter (LPF), yaitu filter yang melalukan sinyal berfrekuensi lebih rendah dari frekuensi cutoff, f_c .
2. High Pass Filter (HPF), yaitu filter yang melalukan sinyal berfrekuensi lebih tinggi dari frekuensi cutoff, f_c .
3. Band Pass Filter (BPF), yaitu filter yang hanya melalukan sinyal yang berfrekuensi antara frekuensi cutoff f_{c1} dan f_{c2} .
4. Band Stop Filter (BSF), yaitu filter yang meredam sinyal yang berfrekuensi antara frekuensi cutoff f_{c1} dan f_{c2} .

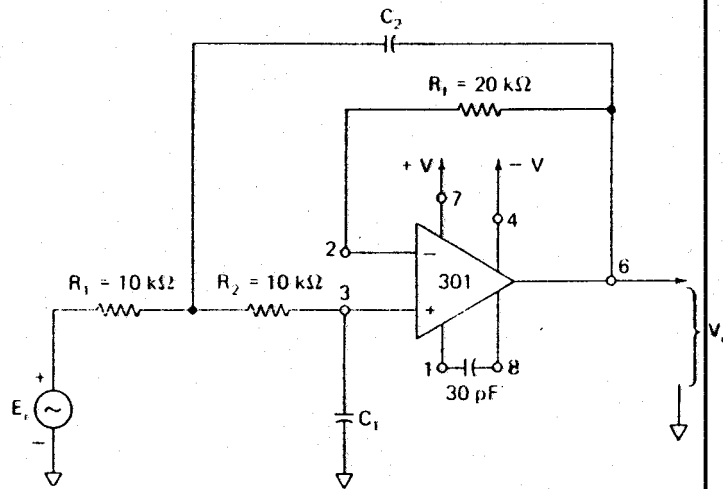
II.4.2.1. Filter Low-Pass

Salah satu filter low-pass adalah filter butterworth -40 dB/decade, rangkaianannya diperlihatkan pada gambar 2.16. Gambar 2.17 menunjukkan tanggapan frekuensinya. Ada empat langkah dalam prosedur perancangan filter ini¹², yaitu:

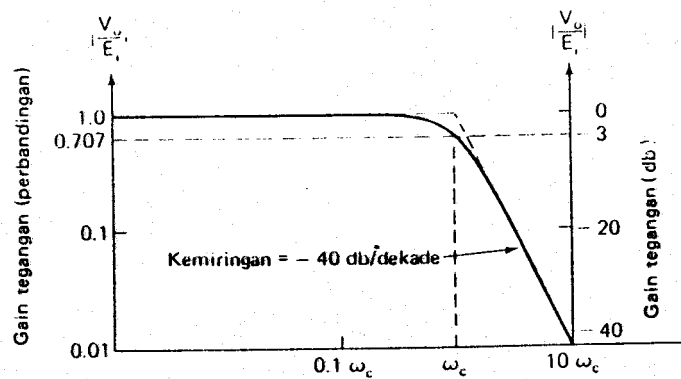
¹¹Robert F. Coughlin, Frederick F. Driscoll, Penguat Operasional dan Rangkaian Terpadu Linear, Erlangga, Jakarta, 1985, hal 260

¹²Ibid, hal 266

1. Pilih frekuensi cutoff, ω_c atau f_c .
2. Buatlah $R_1 = R_2 = R$, dan pilih harga yang sesuai antara 10 dan 100 K Ω . Pilih $R_f = 2R$.
3. Hitunglah C_1 dari $C_1 = 0.707 / (\omega_c * R)$.
4. Pilih $C_2 = 2C_1$.



Gambar 2.16 Filter low-pass -40 dB / decade

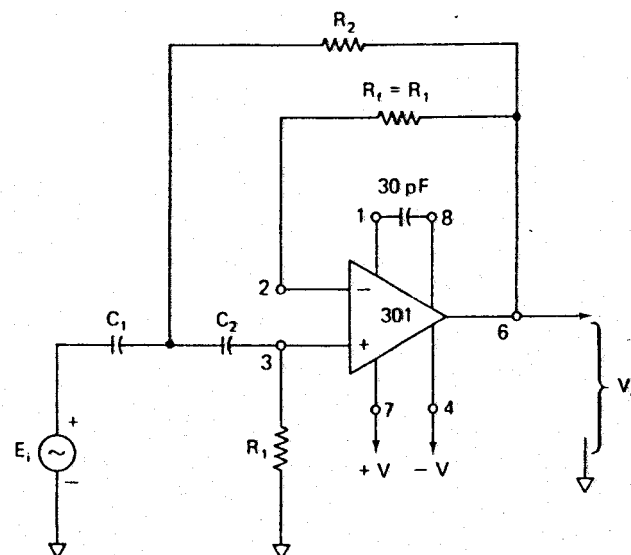


Gambar 2.17 Tanggapan frekuensi untuk filter low-pass -40 dB/dec

II.4.2.2. Filter High-pass

Salah satu filter low-pass adalah filter butterworth 40 dB/decade, rangkaianannya diperlihatkan pada gambar 2.18. Gambar 2.19 menunjukkan tanggapan frekuensinya. Ada empat langkah dalam prosedur perancangan filter ini¹³, yaitu:

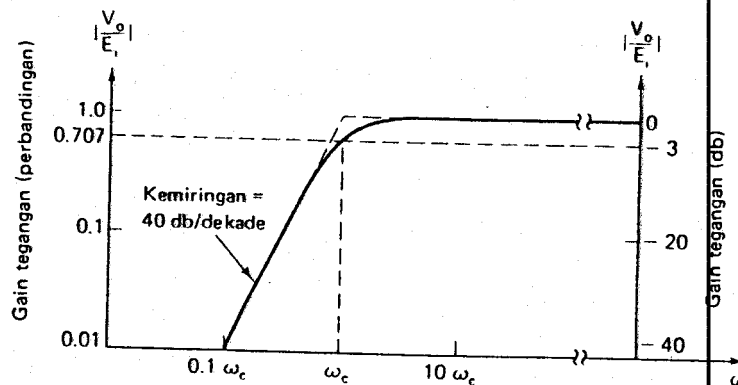
1. Pilih frekuensi cuttoff, ω_c atau f_c .
2. Buatlah $C_1 = C_2 = C$, dan pilih harga yang sesuai.
3. Hitunglah R_1 dari $R_1 = 1.414 / (\omega_c * C)$.
4. Pilih $R_2 = 0.5 R_1$.



Gambar 2.18 Filter high-pass 40 dB / decade

II.4.2.3. Filter Band-Pass

Filter bandpass mempunyai dua frekuensi cut-off, yaitu ω_l dan ω_h . Pita frekuensi antara ω_l dan ω_h adalah lebarpita, B (Bandwidth), jadi $B = \omega_h - \omega_l$. Filter - filter bandpass dapat digolongkan sebagai



Gambar 2.19 Tanggapan frekuensi untuk filter high-pass 40 dB/dec

pita-sempit dan pita-lebar¹⁴. Filter pita-sempit adalah filter yang mempunyai lebar pita lebih kecil dari sepersepuluh frekuensi resonannya ($B < 0.1 \omega_r$). Jika lebarpitanya lebih besar dari sepersepuluh frekuensi resonannya ($B > 0.1 \omega_r$), filtertersebut merupakan sebuah filter pita-lebar. Perbandingan frekuensi resonan terhadap lebarpita dikenal sebagai faktor kualitas, Q , dari rangkaiannya. Q menunjukkan selektifitas rangkaiannya. Makin tinggi harga Q , makin selektif rangkaiannya. Dalam bentuk persamaan $Q = \omega_r / B$ atau $B = \omega_r / Q$ rad/s.

Filter bandpass pita-lebar dapat dibentuk dengan menghubungkan

¹⁴ Ibid, hal 278

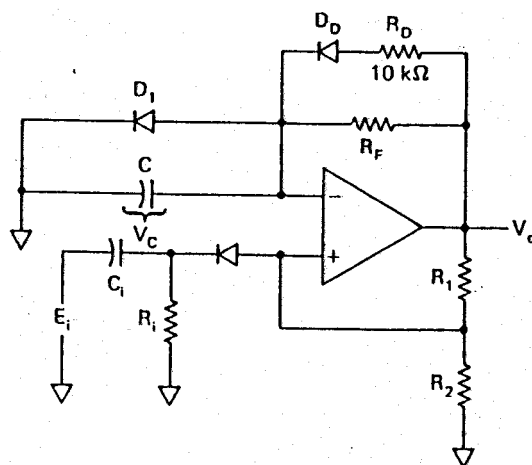
sebuah filter low-pass ke sebuah filter high-pass, namun tidak ada bedanya tentang filter mana yang dilalui sinyal input lebih dahulu.

II.4.3. Multivibrator Monostable

Multivibrator monostable atau multivibrator satu-tembakan membangkitkan sebuah pulsa keluaran tunggal dalam menanggapi sebuah isyarat masukan. Jangka waktu pulsa keluaran itu dapat menjadi lebih lama atau lebih singkat dari isyarat masukannya. Ada beberapa jenis multivibrator satu tembakan ini diantaranya multivibrator satu-tembakan yang tertrigger positif, multivibrator-satu tembakan yang tertrigger negatif, multivibrator satu-tembakan yang tak dapat ditrigger ulang, multivibrator satu tembakan yang dapat ditrigger ulang.

II.4.3.1. Multivibrator Monostable Dengan Op-Amp

Salah satu realisasi multivibrator satu-tembakan yaitu dengan menggunakan operasional amplifier. Gambar 2.20 memperlihatkan



Gambar 2.20 Rangkaian Multivibrator Satu-Tembakan Dengan Op-Amp

rangkaian multivibrator satu-tembakan dengan menggunakan Op-Amp tertrigger negatif.

Ada tiga keadaan dari multivibrator satu-tembakan tersebut, yaitu:

1. Keadaan steady state, yaitu pada saat input trigger nol.
2. Keadaan transisi, yaitu pada saat sinyal trigger masuk.
3. Keadaan penentuan waktu, yaitu lamanya pulsa output dipertahankan.

Rumus yang menunjukkan lamanya keadaan penentuan waktu ini dapat dianalisa, sebagai berikut:

Pada keadaan steady state $V_o = +V_{sat}$, tegangan pada positif input Op-Amp : $V_{ur} = R_z * (+V_{sat}) / (R_1 + R_z)$, tegangan pada negatif input Op-Amp : V_J , yaitu tegangan maju dari diode D_1 . Pada saat trigger negatif masuk maka positif input Op-Amp menjadi negatif (untuk menjamin keberhasilan trigger maka tegangan trigger (V_{tp}) paling tidak besarnya dua kali V_{ur}) dan hal ini menyebabkan positif input Op-Amp lebih negatif dari negatif input Op-Amp, maka output Op-Amp berubah ke $-V_{sat}$ dan tegangan pada positif input Op-Amp menjadi sebesar $V_{Lr} = R_z * (-V_{sat}) / (R_1 + R_z)$, sedangkan tegangan pada negatif Op-Amp bergerak turun dari V_J ke $-V_{sat}$. Secara matematis dapat diuraikan sebagai berikut:

$$V_c(c) = V_J, V_c(\sim) = -V_{sat}.$$

$$V_c(t) = [V_c(o) - V_c(\sim)] * e^{-t/\tau} + V_c(\sim), \quad \tau = R_f * C$$

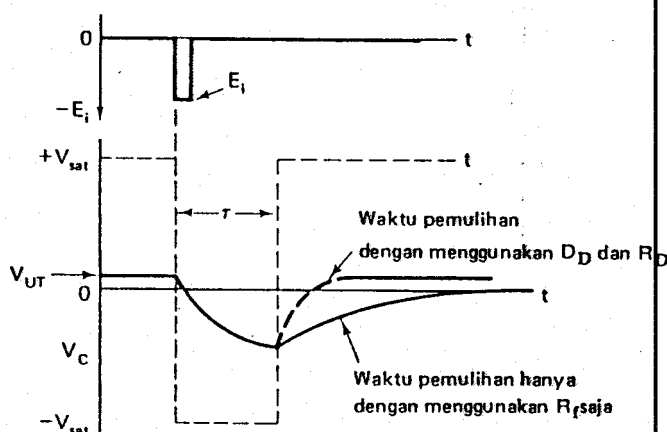
$$V_c(t) = [V_J + V_{sat}] * e^{-t/\tau} - V_{sat} \dots\dots\dots(1)$$

pada saat $t = T$ maka $V_c = V_{Lr}$, T adalah lamanya waktu monostable mempertahankan outputnya. Dengan mensubstitusikan harga V_{Lr} yang dinyatakan dengan harga R_1 dan R_z pada persamaan (1), maka didapat

waktu penentuan T, yaitu:

$$T = R_1 * C \ln \frac{(R_1 + R_2) * (V_J + V_{Sat})}{R_1 * V_{Sat}}$$

Gambar 2.21 memperlihatkan diagram waktu yang menunjukkan keadaan tegangan di C dan output dari monostable (dalam hal ini output Op-Amp). Adanya D_D dan R_D adalah untuk mempersingkat waktu pemulihan (recovery) sehingga multivibrator ini dapat ditrigger lagi dalam selang waktu singkat setelah selang waktu T dari trigger terdahulu, tanpa mempersingkat waktu penentuan T yang dihasilkan oleh trigger berikutnya.



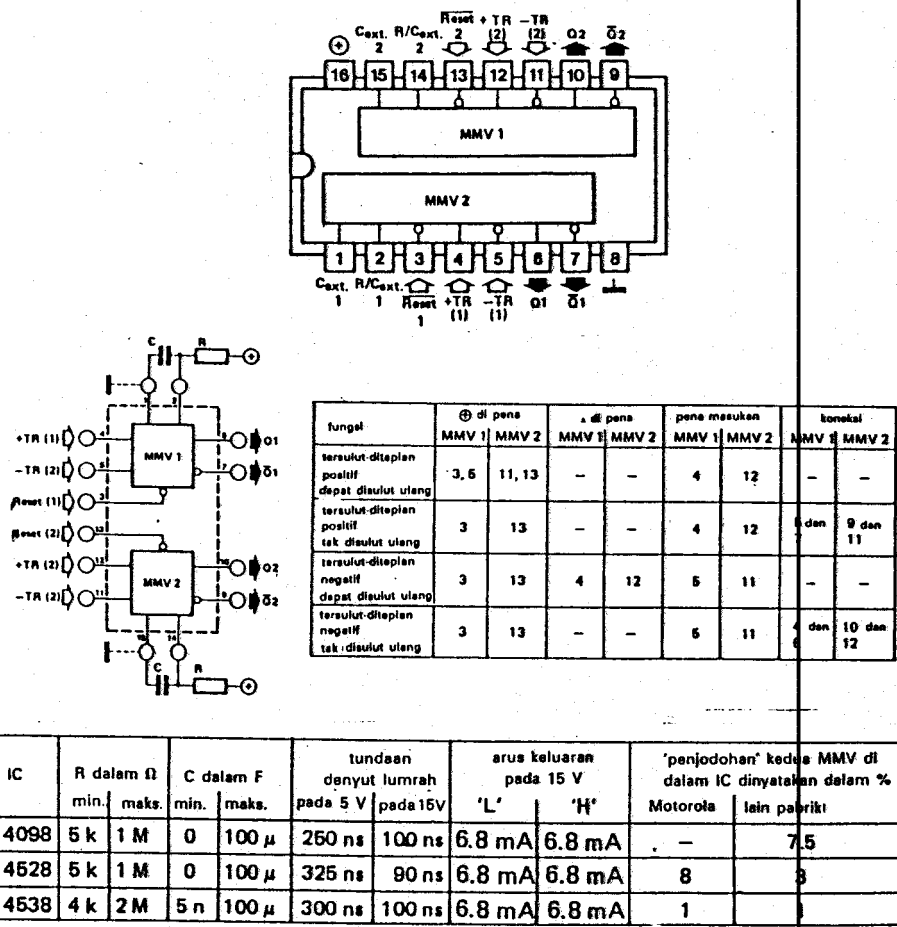
Gambar 2.21 Diagram Waktu Multivibrator Satu-Tembakan

II.4.3.2. Rangkaian Terpadu Multivibrator Monostable

Dewasa ini banyak multivibrator satu-tembakan yang sudah dikemas dalam bentuk rangkaian terpadu, dari tipe CMOS, yaitu IC 4098, 4528, dan 4538, gambar 2.22 memperlihatkan bagian dalam IC tersebut dan koneksi yang diperlukan untuk mewujudkan beberapa fungsi yang dapat

direalisasi serta harga komponen penunjang yang diizinkan¹⁵. Jangka denyut dari multivibrator ini dapat ditentukan dengan menggunakan jaringan RC yang bersesuaian. Jangka denyut $T = R * C$.

Letak pena-pena untuk IC MMV
4098, 4528 dan 4538

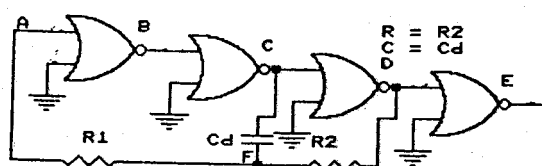


Gambar 2.22 Rangkaian Terpadu Multivibrator Satu-Tembakan

¹⁵ Anonim, Data Praktis Elektronika, PT. Elex Media Komputindo, Jakarta, 1987, hal. 75

II.4.4. Multivibrator Astable

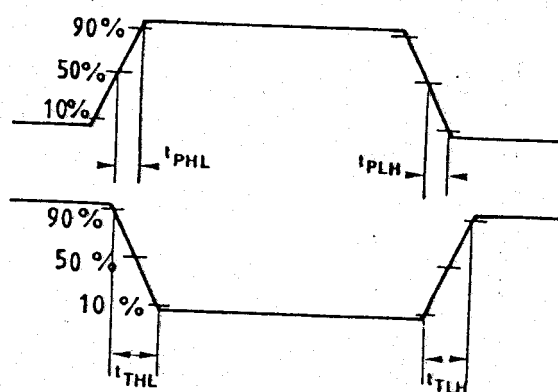
Untuk membangkitkan frekuensi digunakan rangkaian multivibrator astable atau rangkaian clock. Ada beberapa macam realisasinya, misalnya dengan menggunakan Op-Amp atau dengan menggunakan gerbang logika. Besarnya frekuensi yang dibangkitkan tergantung harga R dan C. Gambar 2.23 memperlihatkan salah satu realisasi rangkaian clock dengan gerbang Nor.



Gambar 2.23 Multivibrator Astable Dengan Gerbang Nor

Ada beberapa pengertian tentang gerbang logika, diantaranya:

1. t_{PHL} , t_{PLH} , t_{THL} , t_{TLH} , merupakan waktu tunda propagasi dan waktu transisi. Gambar 2.24 menjelaskan hal ini.

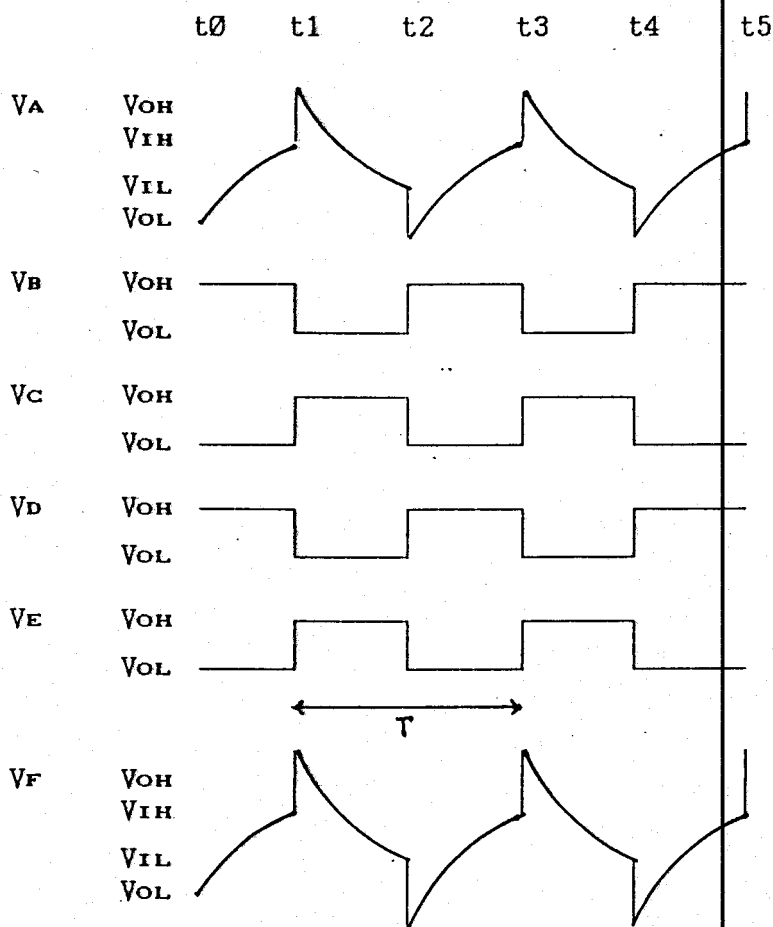


Gambar 2.24 Definisi Waktu Tunda Propagasi Dan Waktu Transisi

2. V_{IH} adalah tegangan input logika 1 minimum.

3. V_{IL} adalah tegangan input logika 0 maksimum.
4. V_{OH} adalah tegangan output logika 1 minimum.
5. V_{OL} adalah tegangan output logika 0 maksimum.

Untuk mendapatkan rumus yang menyatakan besarnya perioda dari rangkaian maka dianalisa menurut timing diagram sebagai berikut:



Gambar 2.25 Timing Diagram Multivibrator Astable Nor Gate

Jika dianalisa mulai dari waktu t_0 sampai t_1 , akan diperoleh:

$$V_F(t) = V_{OL} + (V_{OH} - V_{OL}) * (1 - e^{-t/RC}), R = R_2$$

Sesaat sebelum t_1 , maka $V_F(t) = V_{IH}$, maka diperoleh:

$$t = -RC \ln \left[1 - \frac{V_{IH} - V_{OL}}{V_{OH} - V_{OL}} \right], t \text{ adalah selang waktu antara } t_0 - t_1$$

Sesaat sebelum t_1 , tegangan di Cd, $V_{cd} = V_F - V_C = V_{IH} - V_{OL}$.

Sesaat setelah t_1 , tegangan di titik F, $V_F = V_{OH} + V_{IH} - V_{OL}$.

Analisa antara waktu t_1 sampai t_2 , akan menghasilkan:

$$V_F(t) = [(V_{OH} + V_{IH} - V_{OL}) - V_{OL}] * e^{-t/RC} + V_{OL}$$

Sesaat sebelum t_2 , maka $V_F(t) = V_{IL}$, maka diperoleh:

$$t = -RC \ln \left[\frac{V_{IL} - V_{OL}}{V_{OH} + V_{IH} - 2V_{OL}} \right], t \text{ adalah selang waktu antara } t_1 - t_2$$

Sesaat sebelum t_2 , tegangan di Cd, $V_{cd} = V_F - V_C = V_{IL} - V_{OH}$.

Sesaat setelah t_2 , tegangan di titik F, $V_F = V_{OL} + V_{IL} - V_{OH}$.

Analisa antara waktu t_2 sampai t_3 , akan menghasilkan:

$$V_F(t) = [V_{OH} - (V_{OL} + V_{IL} - V_{OH})] * (1 - e^{-t/RC}) + (V_{OL} + V_{IL} - V_{OH})$$

Sesaat sebelum t_3 , maka $V_F(t) = V_{IH}$, maka diperoleh:

$$t = -RC \ln \left[1 - \frac{V_{IH} - V_{OL} - V_{IL} + V_{OH}}{2V_{OH} - V_{OL} - V_{IL}} \right], t \text{ adalah selang waktu } t_2 - t_3$$

Dari timing diagram terlihat bahwa perioda osilasi adalah sebesar jangka waktu $t_1 - t_2$ ditambah jangka waktu $t_2 - t_3$, ditambah delay digerbang logika Nor, Jadi:

$$T = -RC \ln \left[\frac{V_{IL} - V_{OL}}{V_{OH} + V_{IH} - 2V_{OL}} \right] + -RC \ln \left[1 - \frac{V_{IH} - V_{OL} - V_{IL} + V_{OH}}{2V_{OH} - V_{OL} - V_{IL}} \right] + \text{delay waktu transisi.}$$

Delay waktu transisi selama satu perioda sebesar $2 * 3 * t_{THL} \text{ Nor Gate}$

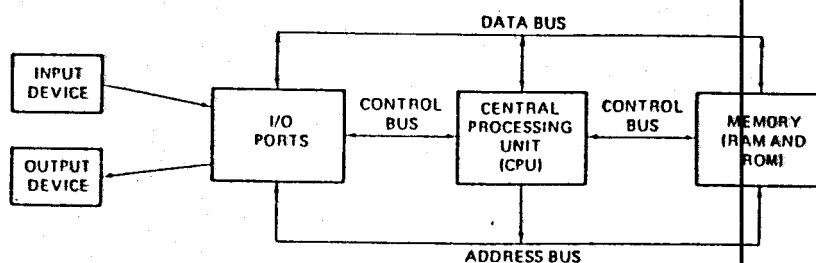
II.4.5. Komparator

Sebuah komparator / pembanding membandingkan tegangan isyarat pada satu masukan dengan dengan masukan lainnya. Komparator dapat direalisasikan dengan Op-Amp, Karena open loop gain Op-Amp yang besar maka perbedaan kecil tegangan masukan diantara kedua masukannya dapat mengubah keadaan tegangan outputnya. Deferensial input E_d yaitu besar tegangan positif input dikurangi dengan besar tegangan negatif

input, menentukan polaritas output dari komparator. Jika E_d positif maka output komparator $+V_{sat}$, jika E_d negatif maka output komparator $-V_{sat}$. Komparator dengan Op-Amp serba guna (seperti 741) mempunyai beberapa kelemahan, diantaranya level tegangan output $+V_{sat}$ dan $-V_{sat}$ yang biasanya sebesar 13 volt, menyebabkan outputnya tak bisa dihubungkan langsung dengan komponen lain seperti IC TTL. Beberapa IC khusus komparator seperti 311, bagian outputnya berfungsi seperti saklar, jika E_d positif maka bagian outputnya seperti saklar terbuka, jika E_d negatif maka bagian outputnya identik dengan saklar tertutup. Dengan menggunakan resistor pull up pada bagian outputnya maka output dapat langsung kompatibel dengan IC TTL. Jika ada gangguan, misalnya noise yang menyebabkan E_d berubah-ubah antara positif dan negatif maka output komparator juga takstabil untuk mengatasi hal ini digunakan umpan balik positif yang menghasilkan histerisis tegangan input sehingga outputnya stabil untuk perubahan E_d yang masih didalam histerisis tegangan inputnya.

II.5. SISTEM MINIMUM 8088 DAN PERIPHERAL PENDUKUNGNYA

Untuk pengaturan sistem pengukuran tekanan darah digunakan mikroprosesor 8088 yang dioperasikan pada mode minimum. Sesuai dengan keperluan maka sistem ini didukung oleh beberapa IC peripheral yaitu PPI 8255, PIT 8253, USART 8251.



Gambar 2.26 Blok Diagram Sistem Minimum

II.5.1. Mikroprosesor 8088

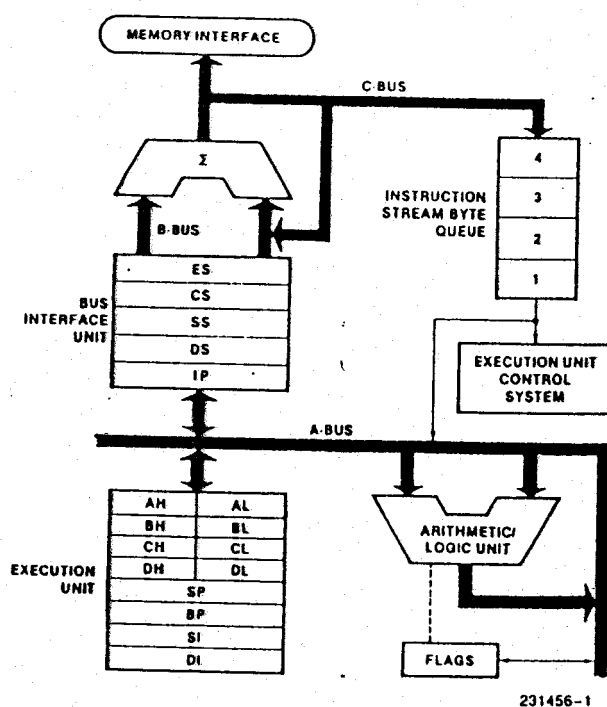


Figure 1. 8088 CPU Functional Block Diagram

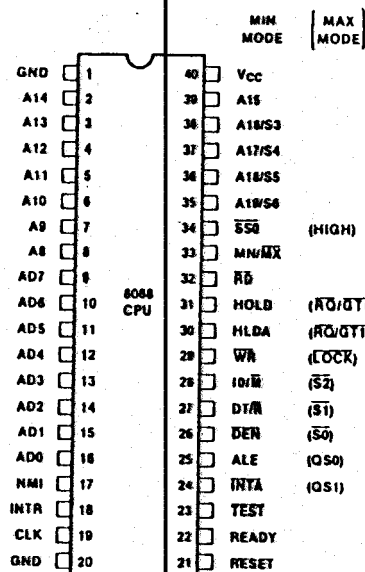


Figure 2. 8088 Pin Configuration

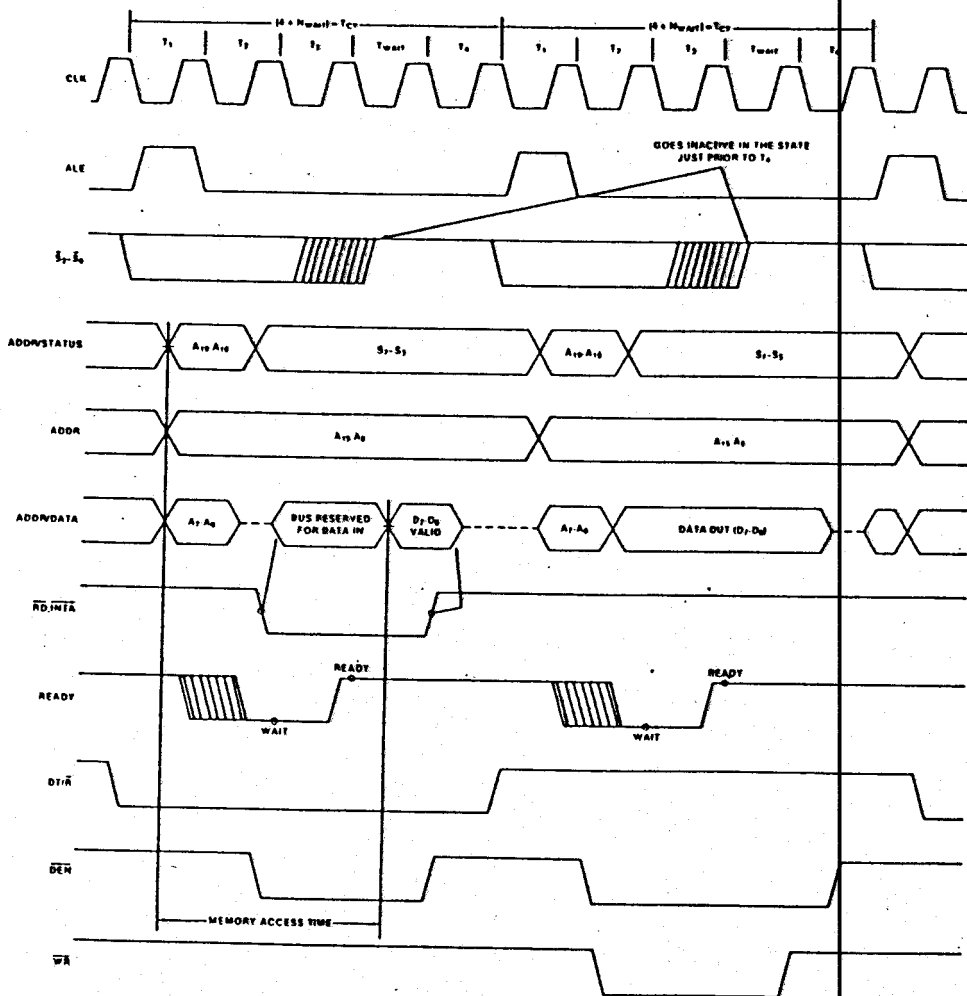
Gambar 2.27 Blok Diagram Fungsi dan Konfigurasi Pin 8088

Gambar 2.27 memperlihatkan blok diagram fungsi dan konfigurasi pin dari CPU 8088.

Mikroprosessor 8088 tersusun atas dua bagian yaitu:

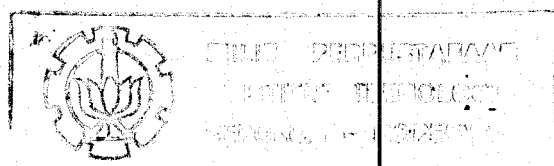
1. Bus Interface Unit (BIU)
2. Execution Unit (EU)

Bus Interface Unit mempunyai beberapa fungsi, diantaranya membangkitkan alamat memori atau I/O untuk transfer data antara



231456-8

Gambar 2.28 Timing Diagram Dari System CPU 8088



peralatan periperal dan 8088, dalam hal ini bagian Execution Unit. Execution Unit menerima kode instruksi program dan data dari BIU, melaksanakan instruksi tersebut, dan menyimpan hasilnya di register dan BIU mengambilnya untuk disimpan ke memori atau dikirim ke I/O. Berikut adalah penjelasan pin-pin pada mikroprosesor 8088 yang digunakan dalam mode minimum¹⁵.

AD0-AD7 (Address Data Bus)

Merupakan jalur address dan data yang bekerja secara multiplex. Jalur ini bekerja sebagai address (A0 - A7) pada bus cycle yang pertama (T1), dan pada bus cycle yang berikutnya (T2, T3 dan T4) jalur ini akan berfungsi sebagai jalur data.

A8 - A15 (Address Bus)

Merupakan address bus untuk seluruh bus cycle (T1 -T4), bus ini tak memerlukan latch oleh ALE agar valid, bus ini akan float / 3 - state off selama interrupt acknowledge atau lokal bus hold acknowledge.

A16/S3 - A19/S6 (Address/ Status)

Jalur ini berfungsi sebagai jalur address dan status yang bekerja secara multiplex. Selama bus cycle yang pertama (T1), jalur ini berfungsi sebagai address (A16 - A19) untuk operasi memory, selama operasi I/O jalur ini akan low. Untuk bus cycle berikutnya jalur ini berfungsi sebagai penunjuk status dari 8088.

¹⁵ Intel Corporation, Microprocessor And Peripheral Handbook Vol. I, Intel Corporation Literature Distribution, Santa Clara, 1987, hal. 2-90

-RD (Read)

RD disebut Read Strobe, bilamana aktif berarti mikroprosesor sedang melaksanakan membaca isi memory atau melakukan pembacaan isi port. Jika $\overline{IO/\bar{M}}$ berlogika 0 berarti sedang membaca memory, tetapi jika $\overline{IO/\bar{M}}$ berlogika 1 berarti membaca port.

READY

Input untuk sinyal dari memori atau I/O yang menunjukkan transfer data segera selesai, sinyal dari memori atau I/O ini disinkronkan dulu oleh clock generator sebelum masuk ke pin READY.

INTR

Sinyal ini merupakan permintaan interupsi ke mikroprosesor 8088 dari perangkat atau komponen di luar.

-TEST

Input ini akan menentukan kerja prosessor jika ada instruksi yang bersifat menunggu untuk suatu uji, jika input ini low maka eksekusi dilanjutkan, jika high maka prosessor akan menunggu dalam idle state.

NMI (Non Maskable Interrupt)

Input untuk menginterrupt 8088, sinyal interrupt harus bersifat edge trigger yang berubah dari low ke high, NMI akan menyebabkan interrupt tipe 2, interrupt jenis ini tak dapat dicegah oleh software, interrupt ini disinkronkan secara internal oleh 8088.

RESET

Dengan adanya signal reset ini, maka mikroprosesor 8088 akan menjalankan instruksi mulai dari awal lagi. Sinyal input harus dipertahankan high untuk paling sedikit 4 siklus clock.

CLK (Clock)

CLK merupakan input clock dari 8088 dan bus controller dan berfungsi untuk mengatur timing dari operasi 8088 dan bus controller. Bentuk pulsa harus memiliki rise time minimal 10 nano detik dan duty cycle 33 %.

VCC

Merupakan pin power supply. Besar tegangan +5 Volt.

GND (Ground)

Merupakan pin ground dari mikroprosesor.

MN/-MX (Minimum/Maximum)

Merupakan pin yang menyatakan mode operasi mikroprosesor 8088. Jika input pin ini berlogika high maka uP 8088 beroperasi dengan mode minimum.

IO/-M

Berfungsi untuk membedakan memory access atau I/O access.

-WR (Write)

Signal ini menyatakan bahwa mikroprosesor 8088 sedang melaksanakan operasi write memory atau I/O tergantung dari pin IO/M. WR akan aktif pada cycle T2, T3 dan Tw dari bus cycle, dan float (3state off) jika lokal bus dalam keadaan hold acknowledge.

-INTA (Interrupt Acknowledge)

Menyatakan bahwa mikroprosesor 8088 telah mendeteksi adanya permintaan interrupt.

ALE (Address Latch Enable)

Merupakan signal output dari 8088 yang digunakan untuk menyimpan address ke dalam address latch.

DT/-R (Data Transmit/Receive)

Diperlukan untuk mengatur arah pengiriman data bus transceiver.

-DEN (Data Enable)

Signal ini berfungsi agar data pada tranceiver dapat di-enable.

HOLD, HLDA

Hold digunakan apabila sebuah prosessor lain ingin mempergunakan bus, dengan memerintahkan ~~8080~~ untuk melepaskan sistem bus, sehingga bus dapat dipakai oleh prosesor lain.

-SSO (Status Line)

Merupakan status line yang bersama-sama dengan IO/M dan DT/R membentuk suatu sistem decoding dari status bus cycle.

Tabel 2.2. Sistem Decoding Dari Status Bus Cycle

IO/-M	DT/-R	-SSO	OPERASI
1	0	0	Interrupt acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0	0	0	Code access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

II.5.2. Programable Peripheral Interface (8255)

IC PPI 8255 merupakan piranti I/O yang luas digunakan, karena mudah pengoperasiannya dan kompatibel dengan berbagai tipe

mikroprosesor, yang berfungsi untuk menghubungkan komponen luar dengan sistem mikroprosesor. Dikatakan port I/O karena disinilah informasi keluar masuk kedalam mikro komputer. Komponen I/O 8255 merupakan paralel I/O dan dapat diprogram untuk mentransfer data dalam berbagai kondisi dari I/O yang sederhana sampai dengan interrupt I/O yang kompleks. 8255 dibentuk dalam 40 pin dual in line package dan dikelompokkan dalam tiga port paralel 8 bit yaitu port A, port B, port C dimana port yang terakhir yaitu port C dapat dipisah dalam 4 bit orde rendah dan 4 bit orde tinggi. Fungsi masing-masing port dapat diprogram sebagai input/output dengan cara menulis control word dalam control register .

Secara garis besarnya operasi PPI 8255 dapat diklasifikasikan dalam 2 mode, yaitu mode bit Set/Reset (BSR) dan mode I/O. BSR digunakan untuk menset/reset bit dalam port, sedang mode I/O dibagi dalam 3 mode yaitu mode 0, mode 1 dan mode 2. Dalam mode 0 semua port berfungsi sebagai I/O yang sederhana, mode 1 adalah mode handshake dan mode 2 dimana peripheral dapat menggunakan port A sebagai port data bi-directional.

Secara blok diagram 8255¹⁶ diperlihatkan dalam gambar 2.29 yang mempunyai dua kelompok 8 bit port A dan B dan kelompok 4 bit port C orde rendah dan orde tinggi, data bus buffer dan control logic.

¹⁶ Douglas V. Hall, Microprocessors And Interfacing Programming And Hardware, McGraw-Hill Book Company, Singapore, 1986, hal.263

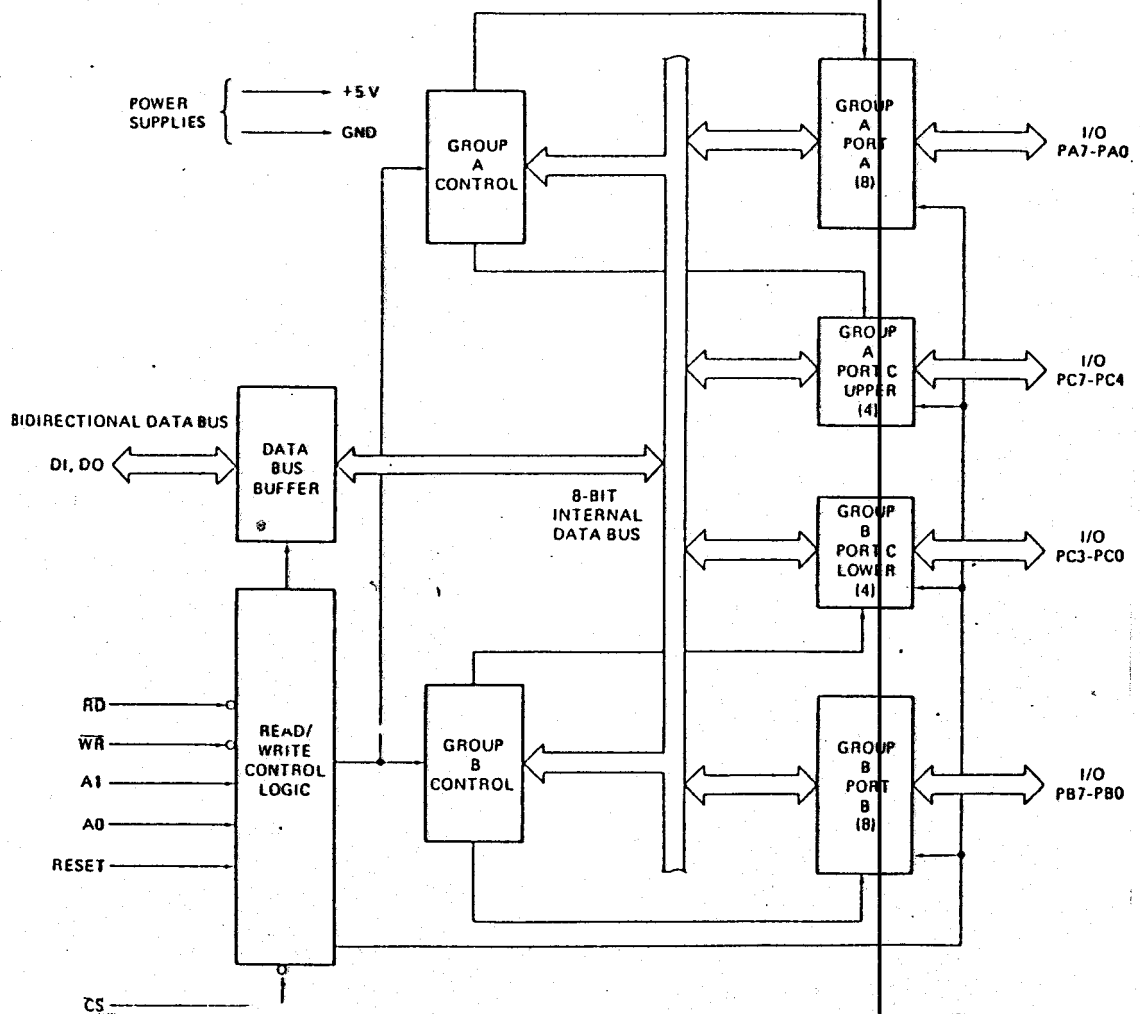
Pada bagian control logic terdapat 6 jalur pengontrol sebagai berikut:

-RD (Read)

Signal kontrol yang memungkinkan operasi baca, dimana jika signal berlogika 0, maka mikroprosesor membaca data pada I/O port.

-WR (Write)

Signal kontrol yang memungkinkan operasi penulisan (write operation) dari I/O port yang dipilih.



Gambar 2.29 Blok Diagram dari PPI 8255

-RESET

Signal aktif high, signal yang akan mereset control register dan men-setting semua port ke dalam mode input.

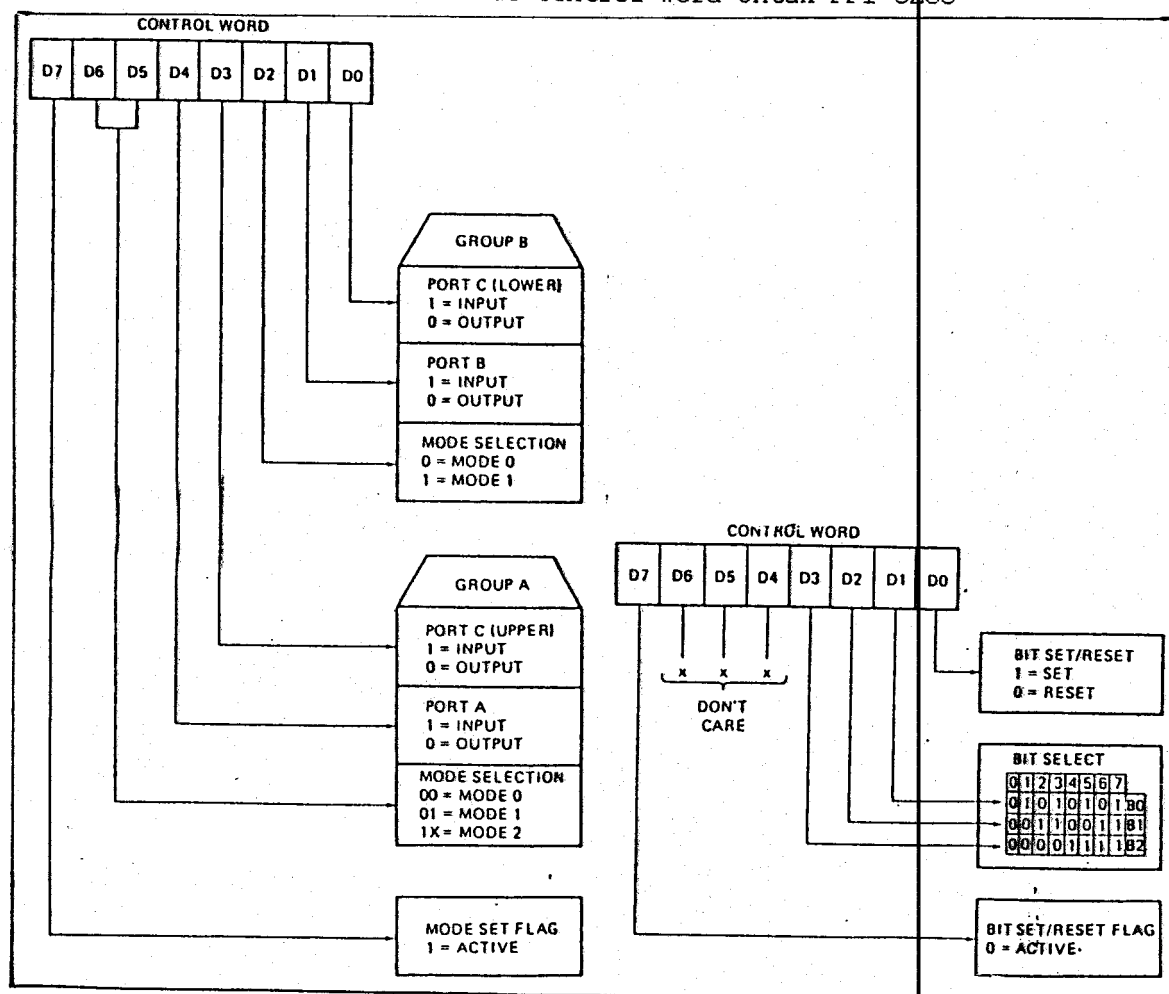
-CS (Chip Select)

Digunakan untuk mengaktifkan chip 8255, bila mendapat logika low mikroprosesor dapat mengakses data ke PPI 8255.

Address Input (A0 - A1)

Kombinasi dari kedua address input ini menentukan register / port dari 8255 yang akan menerima atau mengirim data dari atau ke mikroprosesor

Tabel 2.3. Format Control Word Untuk PPI 8255



Tabel 2.4. Address Decoding Pada PPI 8255

CS	A1	A0	YANG DIPILIH
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	CONTROL WORD REGISTER
1	X	X	8255 Tidak dipilih

Pengoperasian PPI 8255 diawali dengan mengirimkan control word ke alamat control register, control word ini yang akan menentukan mode operasi dari PPI 8255, Gambar 2.30 memperlihatkan format control word untuk mode set-reset dan mode I/O.

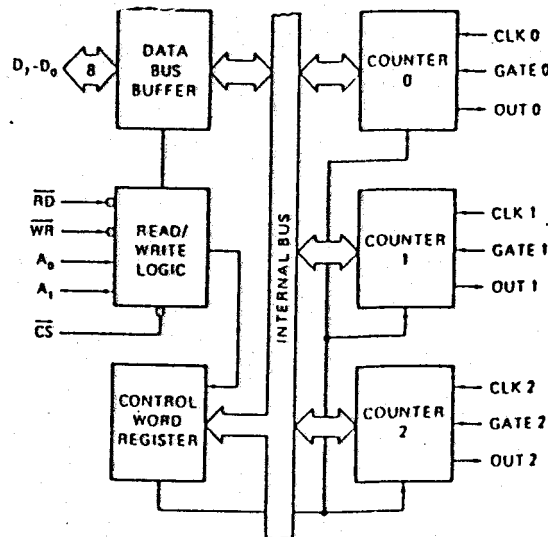
II.5.3. Programmable Interval Timer 8253

Chip 8253 Programable Interval Timer ini dirancang untuk mengatasi persoalan kontrol waktu pada perancangan sistem mikrokomputer. Chip ini terdiri dari tiga buah 16 bit counter yang terpisah satu sama lain dimana setiap counter dapat menerima clock sampai dengan frekuensi 2,6 MHz. Tiap counter dapat diprogram dan diproses dalam 6 mode. Chip ini banyak digunakan sebagai even counter, monostabil multivibrator yang dapat diprogram, timer dan sebagainya. Gambar 2.30 memperlihatkan blok diagram internal PIT 8253, terlihat bahwa PIT 8253 mempunyai 3 buah counter yang dapat diprogram yang berdiri sendiri dan ketiga-tiganya identik. Buffer bus data sudah ada didalamnya sehingga dalam penggunaannya tidak lagi diperlukan penyangga bus data eksternal.

Beberapa pin yang penting dari 8253, yaitu:

Data (D0 - D7)

Bidirectional tri state bus data yang dihubungkan dengan bus data sistem mikroprocessor.



Gambar 2.30 Blok Diagram PIT 8253

Address input (A0, A1)

Input yang digunakan untuk memilih salah satu dari counter atau control word register untuk operasi write atau read. Dihubungkan dengan bus address dari sistem mikroprocessor.

-CS (Chip Select)

Logika low pada input ini akan mengaktifkan tanggapan 8253 terhadap sinyal RD dan WR. Jika diberikan logika high maka 8253 didisable terhadap sinyal RD maupun WR.

-RD (Read)

Control dari proses pembacaan data dari 8253. Input dari sinyal RD yang berlogika low pada operasi read.

-WR (Write)

Input dari sinyal WR yang berlogika nol pada saat proses penulisan data oleh CPU ke 8253.

CLK n

Clock input dari counter n

OUT n

Output dari counter n

GATE n

Input gate dari counter n

Dalam sistem mikrokomputer berbasis mikroprocessor 8088, PIT diperlakukan sebagai peralatan peripheral I/O. Dalam pengoperasiannya untuk menentukan mode kerja tiap counter, maka sebelum counter dioperasikan harus diawali dulu dengan mengirimkan control word ke control word register, hal ini dilakukan secara terpisah untuk masing-masing counter, masing-masing counter mempunyai alamat tertentu yang ditentukan oleh A1 dan A0, serta -CS.

Tabel 2.5. Address Decoding Pada PIT 8253

CS	A1	A0	YANG DIPILIH
0	0	0	COUNTER 0
0	0	1	COUNTER 1
0	1	0	COUNTER 2
0	1	1	CONTROL WORD REGISTER
1	X	X	8253 Tidak dipilih

Tabel 2.6. Format Control Word PIT 8253

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Tabel 2.7. Uraian Format Control Word PIT 8253

Pilih Counter

SC1	SC0	COUNTER
0	0	0
0	1	1
1	0	2
1	1	Illegal

Read Write

RL1	RL0	OPERASI
0	0	Mengunci buffer counter
0	1	Read/Load byte rendah
1	0	Read/Load byte tinggi
1	1	Read/Load byte tinggi kemudian byte rendah

Mode

M2	M1	M0	MODE
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

BCD

BCD	OPERASI COUNTER
0	Biner 16 bit
0	BCD 4 decade

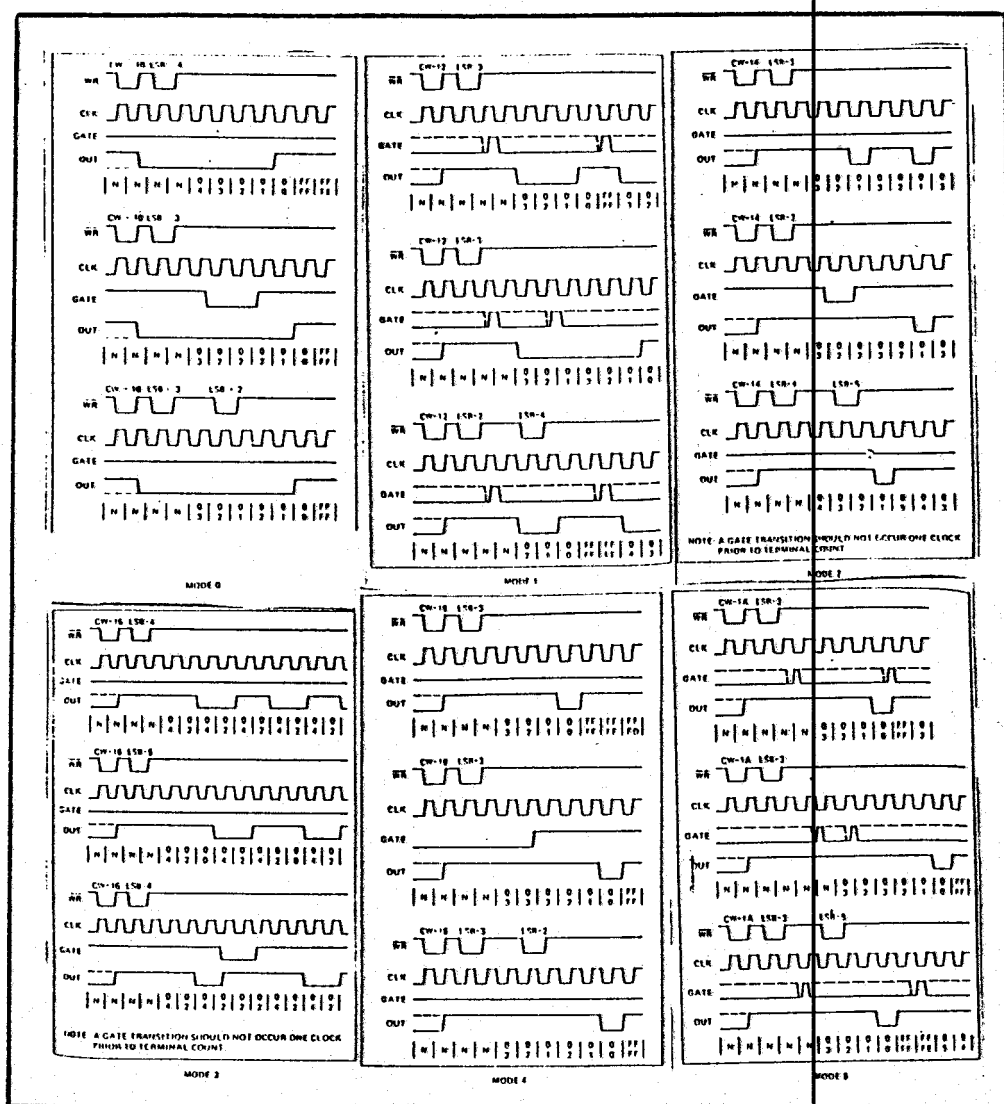
Programmable Interval Timer (8253) dapat dioperasikan dalam 6 mode.

Adapun mode-mode itu adalah sebagai berikut:

1. MODE 0: Interrupt pada Terminal Count.
2. MODE 1: Hardware_Retriggerable Single-Shot.
3. MODE 2: Rate Generator.
4. MODE 3: Square Wave Mode.
5. MODE 4: Software_Triggered Strobe.
6. MODE 5: Hardware_Triggered Strobe.

Gambar 2.31 memperlihatkan timing diagram dari PIT 8253 untuk masing-masing mode.

Untuk mengetahui isi counter pada saat tertentu dimana counter masih beroperasi, tanpa mempengaruhi operasi counter, maka perintah counter latch / mengunci buffer counter dilaksanakan, caranya dengan mengirim



Gambar 2.31 Timing Diagram 8253 Untuk Masing-masing Mode

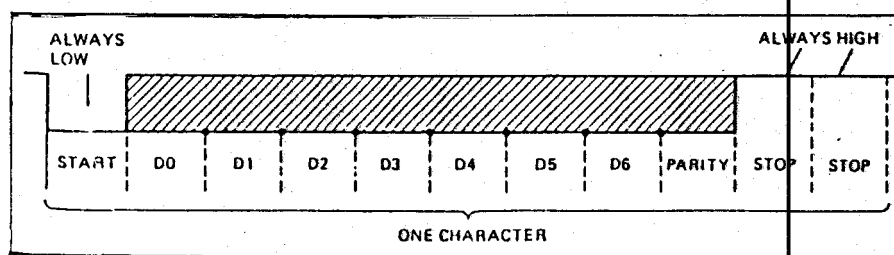
control word, dimana pada posisi RL1 dan RL0 diberi nilai 0, posisi SC1 dan SC2 diisi sesuai dengan counter yang akan diketahui isinya, sedangkan 4 bit yang lainnya diisi dengan nol. Setelah control word tersebut dikirim ke alamat Control Word Register maka dilakukan pembacaan data pada alamat counter yang ingin diketahui isinya.

II.5.4. Universal Synchronous Asynchronous Receiver Transmitter

(USART) 8251

Pada sistem mikrokomputer, data diolah secara paralel. Untuk transfer jarak jauh, data paralel kurang efisien karena membutuhkan jumlah kabel yang banyak, oleh karenanya data paralel harus dikonversikan ke dalam bentuk serial. Metoda transmisi data serial dapat berupa asinkron maupun sinkron. Pada transmisi sinkron data dikirim dalam blok dengan kecepatan tetap, awal dan akhir dari blok diidentifikasi dengan byte atau bit tertentu. Pada transmisi asinkron setiap karakter data terdiri dari 1 yang mengidentifikasi awal karakter dan 1 atau 2 bit sebagai akhir karakter, karena setiap karakter diidentifikasi sendiri-sendiri maka karakter dapat dikirim pada setiap saat (asinkron).

Tabel 2.8. Format Bit Transmisi Serial Asinkron



Saat tak ada data yang dikirim, sinyal line dalam keadaan logik 1 atau mark. Awal karakter ditandai dengan sinyal line menjadi logik 0 atau space selama 1 waktu bit (bit time), bit data kemudian dikirim sesudahnya mulai dengan bit orde terendah (LSB), lebar data dapat terdiri dari 5, 6, 7, atau 8 bit. Sesudah data bit adalah parity bit yang dapat berupa parity genap atau ganjil atau tanpa parity, parity digunakan untuk memeriksa ada tidaknya kesalahan pada data yang diterima. Sesudah data bit dan parity bit maka line harus kembali ke logik 1 atau mark paling sedikit 1 waktu bit (bit time). bit terakhir ini dinamakan stop bit, stop bit dapat berupa 1 atau 2 bit. Baud Rate atau kecepatan transfer dari data serial didefinisikan sebagai $1 / (\text{waktu 1 bit})^{17}$. Jika waktu bit 3.33 ms maka Baud Rate nya $1 / 3.33 \text{ ms}$ atau sebesar 300 Baud. Baud rate yang sering digunakan yaitu 110, 300, 1200, 2400, 4800, 9600, dan 19200 Baud.

Untuk menghubungkan mikrokomputer dengan line data serial, data harus dikonversikan menjadi data serial, konversi dapat berupa paralel in - serial out dan serial in - paralel out register geser. Untuk beberapa peralatan membutuhkan rangkaian handshaking agar peralatan pengirim tidak mengirim data lebih cepat dari penerimaan data pada penerima. Komponen yang dibutuhkan untuk komunikasi serial data asinkron atau sinkron dinamakan USART. USART 8251A adalah komponen yang dibuat oleh INTEL dengan teknologi N-Channel Silikon gate, untuk mengimplementasikan konversi data paralel ke serial atau

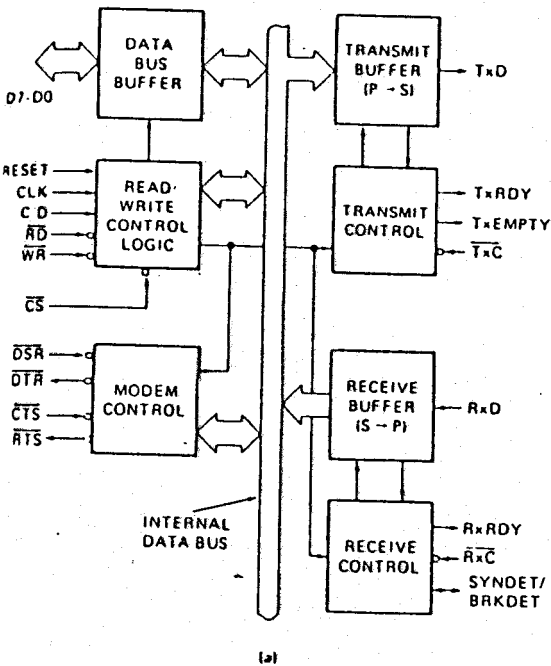
¹⁷ Ibid, hal. 443

sebaliknya. USART 8251A kompatibel dengan keluarga mikroprosessor INTEL, seperti MCS-48 dan IAPX-86,88. CPU dapat membaca lengkap status USART pada setiap saat, termasuk kesalahan transmisi data dan sinyal kontrol seperti SYNDET dan TxEMPTY, sesuai dengan namanya maka USART dapat digunakan untuk komunikasi data sinkron maupun asinkron. Gambar 2.32 menunjukkan blok diagram dan fungsi pin dari USART 8251A.

Fungsi masing masing bloknnya adalah sebagai berikut:

1. Data Bus Buffer

Buffer 8 bit yang bidirectional ini menghubungkan 8251A dengan sistem data bus dari mikrokomputer, di sini disimpan data paralel 8 bit yang akan dikeluarkan ke jalur data, maupun yang diterima dari jalur data. Data dikirim dan diterima dengan instruksi OUT dan IN dari



8251 Pin Functions

Pin Name	Pin Function
D7-D0	Data bus (8 bits)
C/D	Control or data as to be written or read
RD	Read data command
WR	Write data or control command
CS	Chip select
CLK	Clock pulse (TTC)
RESET	Reset
TxC	Transmitter clock
TxD	Transmitter data
RxC	Receiver clock
RxD	Receiver data
RxRDY	Receiver ready (has character for CPU)
TxRDY	Transmitter ready (ready for char. from CPU)
DSR	Data set ready
DTR	Data terminal ready
SYNDET/BD	Sync detect/break detect
RTS	Request to send data
CTS	Clear to send data
TxEMPTY	Transmitter empty
Vcc	+5-V supply
GND	Ground

Gambar 2.32 Blok Diagram Dan Pin-pin USART 8251A

CPU. Di samping itu untuk mode word dan command word serta informasi status juga ditransfer melalui buffer data bus ini.

2. Read / Write Control Logic

Blok ini berfungsi untuk mengendalikan perpindahan data dan common status, sehingga memungkinkan data dapat dibaca maupun ditulis pada lokasi dan waktu yang tepat.

3. Modem Control

Bagian ini digunakan sebagai perantara hubungan antara 8251A dengan modem.

4. Receiver Buffer (S-P) Dan Receiver Control

Dua blok ini bekerja bersama-sama dalam menerima data serial. Receiver buffer akan mengubah data serial menjadi data paralel agar dapat dibaca CPU, sedangkan receiver control memantau status dari receiver buffer.

5. Transmit Buffer (P-S) Dan Transmit Control

Dua blok ini bekerja bersama-sama dalam mentransmisikan data serial. Transmit buffer berfungsi menerima data dari CPU, kemudian secara otomatis pada data ditambahkan bit start, bit paritas, dan stop bit sesuai dengan mode word yang diberikan. Setelah diletakkan pada buffer register pengirim, data yang telah dibingkai tadi digeser keluar menjadi deret data berbentuk serial yang siap dikirim, sedangkan transmit control memantau status dari transmit bufffer.

USART 8251A dikemas dalam bentuk chip 28 pin DIP, beberapa fungsi pinnya sebagai berikut:

DO - D7

Merupakan jalur data 8 bit yang digunakan untuk hubungan dengan jalur

data sistem mikroprosessor. Semua informasi termasuk inisialisasi pemrograman word register dilewatkan melalui jalur ini.

RESET

Logika high pada pin ini menyebabkan 8251A dalam keadaan idle. Pulsa reset minimum 6 perioda pulsa clocknya.

Clock

Untuk mensinkronisasi semua operasi dalam USART. Clock di sini bukan clock untuk menentukan Baud Rate. Frekuensi clock minimum 30 kali bit rate pengiriman data.

-WR (Write)

Input low pada pin ini menyebabkan CPU menuliskan data atau control word ke USART.

-RD (Read)

Input low pada pin ini menyebabkan CPU membaca data atau status dari USART 8251A.

C/-D

Masukkan pada pin ini berfungsi memberi informasi pada USART 8251A, jenis data yang ada pada data bus, data atau control word.

-CS (Chip Select)

USART 8251A diaktifkan dengan memberikan logika low pada pin ini. Logika high akan menyebabkan bus data dalam keadaan mengambang.

-DSR (Data Set Ready)

Pin masukkan ini dapat dideteksi oleh CPU. Biasanya digunakan untuk menguji keadaan modem.

-DTR (Data Terminal Ready)

Pin keluaran ini biasanya digunakan untuk data terminal ready.

-CTS (Clear To Send)

Logika low pada pin ini menyebabkan 8251A dapat mengirim data serial, jika bit TxEn pada command word diaktifkan, pin ini juga diperlukan untuk 'handshaking'.

-RTS (Request To Send)

Pin keluaran ini multiguna, biasanya dipakai untuk 'handshaking'.

TxD (Transmit Data)

Pin ini merupakan jalur keluaran bagi data serial yang akan dikirimkan.

RxD (Receive Data)

Pin ini merupakan jalur masukan, dimana data serial diterima.

TxC (Transmitter Clock)

Pin masukan bagi clock yang diperlukan untuk mengendalikan baud rate pengiriman. Pada modus transmisi sinkron nilai baud rate sama dengan frekuensi dari TxC. Pada modus transmisi asinkron, baud ratenya merupakan faktor perkalian dari frekuensi TxC. Faktor perkalian tersebut dapat 1, 1/16, atau 1/64 kali frekuensi TxC. Misalnya bila kita ingin mengirim data pada 300 baud, maka:

TxC sama dengan 300 Hz pada mode 1x

TxC sama dengan 4800 Hz pada mode 16x

falling edge dari TxC menggeser data serial keluar dari 8251A.

RxC (Receiver Clock)

Pin masukan bagi clock yang diperlukan untuk mengendalikan baud rate penerimaan data. Pada modus transmisi sinkron nilai baud rate sama dengan frekuensi RxC. Pada modus transmisi asinkron, baud ratenya merupakan faktor perkalian dari frekuensi RxC. Faktor perkalian

tersebut dapat berupa 1, 1/16, atau 1/64 kali frekuensi RxC. Data Serial akan disampel 8251A pada rising edge RxC.

TxRDY (Transmitter Ready)

Untuk memberitahu CPU bahwa pengirim telah siap menerima data untuk dikirimkan.

RxRDY (Receiver Ready)

Untuk memberitahu bahwa pada penerima telah siap dibaca CPU. Kondisi pin ini dapat dideteksi menggunakan operasi pembacaan status.

Syndet/BD (Syn Detect / Break Detect)

Mempunyai dua fungsi:

- Pada mode sinkron, bila input data RxD berada dalam keadaan '0' selama lebih dari dua waktu karakter, pin akan high.
- Pada mode asinkron pin ini akan high bila 8251A mendeteksi adanya karakter sinkronisasi pada deret bit data.

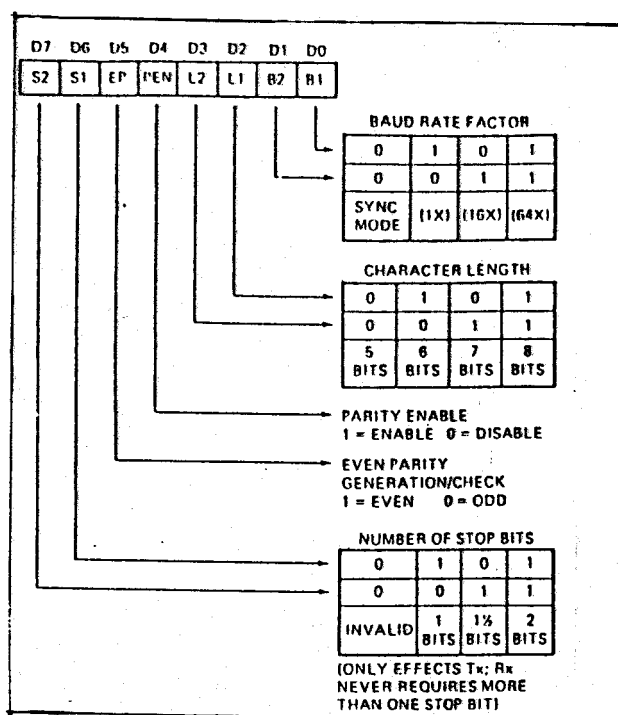
Sebelum melakukan fungsinya USART 8251A harus diprogram dulu dengan jalan menginisialisasi. Inisialisasi dilakukan oleh CPU dengan jalan mengirim control word ke alamat control register. Operasi 8251A ditentukan oleh nilai yang diberikan pada bit-bit control word, yang dibagi dalam dua format, yaitu:

1. Mode Instruction
2. Command Instruction

Untuk menginisiasinya, maka yang pertama dikirim adalah mode instruction, mode instruction cukup sekali ditulis setelah operasi reset, mode instruction mendefinisikan karakteristik operasi 8251A. Gambar 2.36 memperlihatkan format mode instruction. Setelah pengiriman mode instruction maka dilakukan pengiriman command instruction,

command instruction akan mengontrol operasi 8251A. Untuk memeriksa apakah terjadi kesalahan atau ada kondisi lain yang perlu ditangani

Tabel 2.9. Format Mode Instruction 8251A

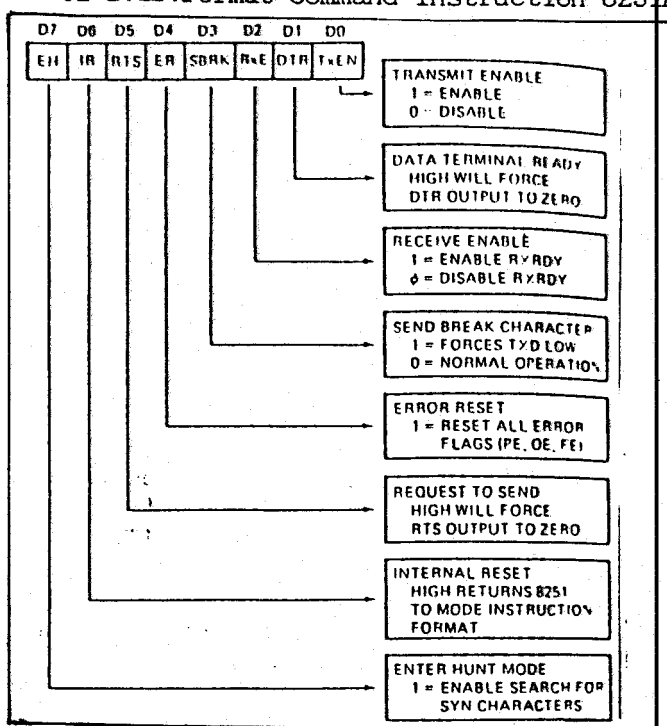


CPU maka 8251A menyediakan fasilitas pembacaan status, yang dapat dilakukan setiap saat selama operasinya. Beberapa bit dari format pembacaan status mempunyai arti yang sama dengan pin keluaran, hal ini memungkinkan 8251A dapat digunakan pada sistem polling.

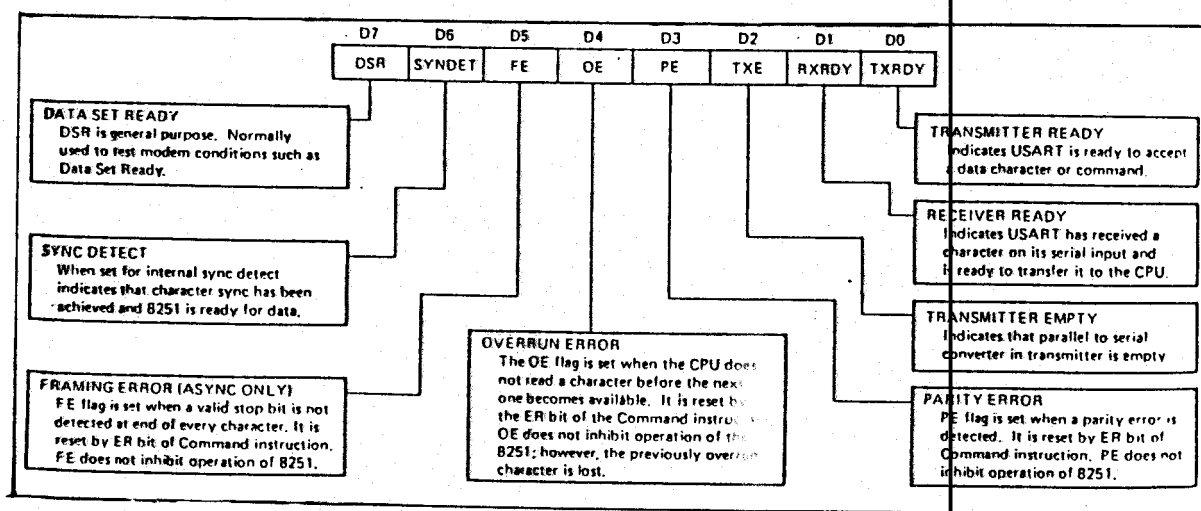
Untuk mengoperasikan USART 8251A pada modus asinkron, urutan pemrograman yang harus dilakukan adalah:

1. Memberikan reset pada 8251A, bisa secara eksternal maupun internal lewat software.
2. Menuliskan mode instruction pada control register.

Tabel 2.10. Format Command Instruction 8251A



Tabel 2.11. Format Status Word 8251A



3. Menuliskan command instruction pada control register.

Perintah reset digunakan untuk menginisialisasi urutan pemrograman.

Perintah berikutnya otomatis dianggap sebagai mode instruction.

Setelah itu data yang ditulis pada control register dianggap sebagai command instruction. Untuk mengubah mode instruction harus dilakukan reset, dengan cara memberikan pulsa reset atau menuliskan command instruction dengan bit-6 diset high.

Untuk mengoperasikan USART 8251A pada modus sinkron, urutan pemrograman yang harus dilakukan adalah:

1. Memberikan reset pada 8251A, bisa secara eksternal maupun internal lewat software.
2. Menuliskan mode instruction pada control register.
3. Menuliskan 1 atau 2 karakter sinkronisasi pada control register
4. Menuliskan command instruction pada control register.

Seperti pada modus asinkron, mode instruction hanya dapat ditulis setelah dilakukan reset terlebih dahulu.

II.5.5. Metoda Dan Standar Transmisi Serial Asinkron

Pada dasarnya metoda transmisi serial mempunyai 2 macam aturan atau protokol perangkat keras¹⁸, yaitu:

1. Current Loop 20mA

Aturan ini merupakan sistem lama dan menggunakan arus, yaitu adanya arus mengalir menandakan keadaan idle atau keadaan istirahat. Pengiriman informasi terjadi bila arus berubah besarnya. Arus minimum sebesar 18 mA harus ada dan arus maksimum besarnya 25 mA untuk menjaga agar disipasi tidak berlebihan. Pada mula sistem ini dipakai untuk

¹⁸ Ibid, hal. 448

mentransfer data biner serial, misalnya teleprinter. Keuntungan sistem ini adalah sederhana, ekonomis, dan murah. Sedangkan kerugiannya noise filtering sukar dan tidak ada pembakuan tegangan, serta membangkitkan cross-talk bila kabel panjang.

2. RS-232C

Semua sinyal listrik yang disalurkan pada media transmisi akan bertambah lemah karena adanya hambatan, kapasitansi, serta induktansi media transmisi itu sendiri. Akibatnya sinyal informasi dapat terganggu oleh derau. Penyaluran informasi ke tempat yang jaraknya jauh membutuhkan penguatan supaya informasi dapat tiba di tempat tujuan, supaya peralatan komunikasi dapat berhubungan dibuat ketentuan baku, yaitu Data Terminal Equipment dan Data Communication Equipment yang melakukan pertukaran data secara serial dengan sinyal biner yang dapat menggunakan standar RS-232C. Standar ini berisi karakteristik sinyal listrik, karakteristik mekanik, dan cara operasi rangkaian fungsional.

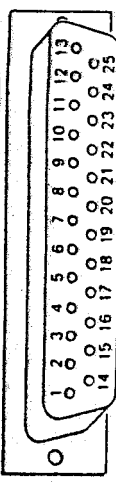
Beberapa karakteristik sinyal listrik RS-232C:

1. Tegangan rangkaian terbuka maksimum 25 V.
 - Keadaan Logika 1 (mark) ditandai dengan tegangan yang lebih negatif dari -3 V. Pada logika 1 besarnya tegangan harus antara -5 V dan - 15 V.
 - Keadaan logika 0 (space) ditandai dengan tegangan yang lebih positif dari + 3 V. Pada logika 0 besarnya tegangan harus antara + 5 V dan + 15 V.
2. Slew rate (perubahan tegangan keluaran per satuan waktu) harus lebih kecil dari 30 V/mikro-detik. Waktu yang diperlukan untuk

melewati daerah peralihan - 3 V sampai + 3 V tidak boleh lebih dari 1 milidetik.

Karakteristik mekanik untuk antarmuka baku ini juga ditentukan yaitu penggunaan konektor DB25P dan DB25S. Tabel 2.11. menunjukkan DB25P, nama pin, dan sinyal-sinyalnya.

Tabel 2.12. DB-25P Connector Dan Nama Pin Serta Sinyal-sinyalnya



PIN NUMBER	COMMON NAME	RS 232-C NAME	DESCRIPTION	SIGNAL DIRECTION ON DCE
1		AA	PROTECTIVE GROUND	-
2	TXD	1A	TRANSMITTED DATA	IN
3	RXD	1B	RECEIVED DATA	OUT
4	RTS	CA	REQUEST TO SEND	IN
5	CTS	12	CLEAR TO SEND	OUT
6	DSR	CC	DATA SET READY	OUT
7	GND	AB	SIGNAL GROUND (COMMON RETURN)	-
8	CD	CF	RECEIVED LINE SIGNAL DETECTOR	OUT
9		-	(RESERVED FOR DATA SET TESTING)	-
10		-	(RESERVED FOR DATA SET TESTING)	-
11		SCF	UNASSIGNED	-
12		SCB	SECONDARY REC'D. LINE SIG. DETECTOR	OUT
13		SBA	SECONDARY CLEAR TO SEND	OUT
14		1B	SECONDARY TRANSMITTED DATA	IN
15			TRANSMISSION SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
16		SBB	SECONDARY RECEIVED DATA	OUT
17		DD	RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
18			UNASSIGNED	-
19		SCA	SECONDARY REQUEST TO SEND	IN
20	DTR	CD	DATA TERMINAL READY	IN
21		CG	SIGNAL QUALITY DETECTOR	OUT
22		CE	RING INDICATOR	OUT
23		CH/CI	DATA SIGNAL RATE SELECTOR (DTE/DCE SOURCE)	IN/OUT
24		DA	TRANSMIT SIGNAL ELEMENT TIMING (DTE SOURCE)	IN
25			UNASSIGNED	-

Protective Ground

Pin ini dapat dihubungkan secara listrik dengan frame dari pada peralatan yang kemudian dapat dihubungkan dengan tanah kalau diperlukan.

Sinyal Ground

Sebagai potensial acuan untuk semua rangkaian kecuali protective

ground.

Transmitted Data

Sinyal ini dibangkitkan oleh DTE diberikan pada DCE untuk disalurkan melalui saluran komunikasi pada tujuannya. Ketentuan RS-232C meminta agar DTE harus dalam keadaan mark saat tidak ada data yang dikirim atau pada interval antar karakter. Transmisi melalui saluran komunikasi hanya dapat dilaksanakan DTE bila dalam keadaan ON (space) ditemui pada empat rangkaian berikut: RTS, CTS, DSR, dan DTR.

Received Data

Sinyal data yang dibangkitkan oleh DCE berdasarkan sinyal yang diterimanya, sinyal ini diberikan ke DTE, RS-232C menghendaki sinyal ini harus dalam keadaan mark bila sinyal carrier detect dalam keadaan off.

Request to Send

Sinyal ini mengendalikan DCE lokal bagian transmit guna menyalurkan data. Keadaan on menyebabkan rangkaian DCE siap mengirim data. DTE kalau hendak mengirim data akan membuat rangkaian RTS ini on. Untuk menyatakan bahwa DCE telah siap, rangkaian Clear to Send menjadi on, sehingga DTE dapat mengirimkan datanya melalui transmit data.

Clear to Send

Sinyal ini menyatakan bahwa DCE siap data melalui saluran komunikasi. Keadaan off (mark) pada rangkaian ini menyatakan DTE untuk tidak mengirimkan data.

Data Set Ready

Untuk menyatakan status dari modem yang terhubung pada DCE. On

menyatakan bahwa peralatan tersebut telah terhubung pada saluran dan DCE lokal siap bertukar sinyal kendali dengan DTE untuk memulai pertukaran data.

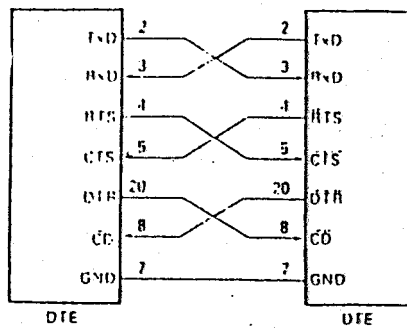
Carrier Detect

Status on menyatakan bahwa saluran dalam keadaan baik dan dapat menerima maupun mengirimkan data. Status off menyatakan bahwa saluran dalam keadaan kurang baik sehingga tidak dapat mengadakan demodulasi.

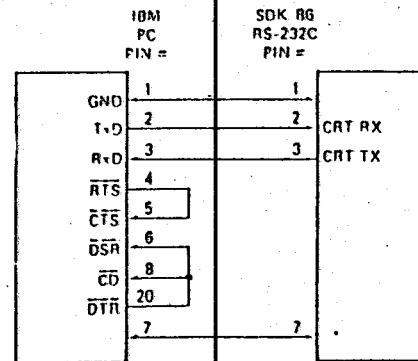
Hubungan antara dua komputer dimana jaraknya dekat, tidak memerlukan modem, namun tetap mengikuti standar RS-232C, hubungan ini dinamakan hubungan cross-over atau sering disebut null modem. Gambar 2.33 memperlihatkan hubungan tersebut.

Bila komputer dihubungkan dengan peralatan lain maka hubungan ini dapat menggunakan sebagian, seluruh, atau tanpa sinyal handshake, card IBM PC-XT asinkron dikonfigurasi sebagai DTE. Supaya card IBM PC-XT asinkron mampu berkomunikasi, maka input CTS, DSR, Cd harus diaktifkan. Bios akan mengaktifkan output dari rangkaian DTR dan RTS. Untuk card tersebut maka RTS pada pin 4 dihubungkan ke CTS pada pin 5, sehingga CTS akan aktif begitu RTS dihidupkan. Pin 6 dan pin 8, serta pin 20 juga dihubungkan supaya saat IBM PC-XT mengaktifkan output DTR, input DSR dan input CD akan aktif. Gambar 2.39 memperlihatkan hal ini.

Salah satu cara untuk konversi antara level RS-232C dan TTL adalah dengan menggunakan IC MC 1488 berupa driver dari TTL ke RS-232C dan MC 1489 berupa receiver dari RS-232C ke TTL, seperti ditunjukkan gambar 2.34 MC 1488 membutuhkan catu positif dan negatif. Sedangkan MC 1489 hanya membutuhkan +5 V. Kapasitor dari output dari MC 1488 untuk mengurangi cross-talk antara kabel yang berdekatan, juga rise

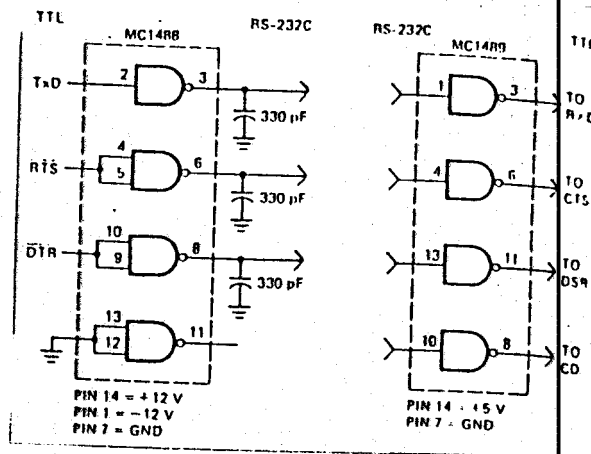


(a)



(b)

Gambar 2.33 (a) Hubungan Null Modem (b) Hubungan RS-232C Tanpa Sinyal Handshaking



Gambar 2.34 Interfacing Antara TTL dan RS-232C

time dan fall time dari sinyal RS-232C harus dibatasi menjadi 30 V/mikro-detik.

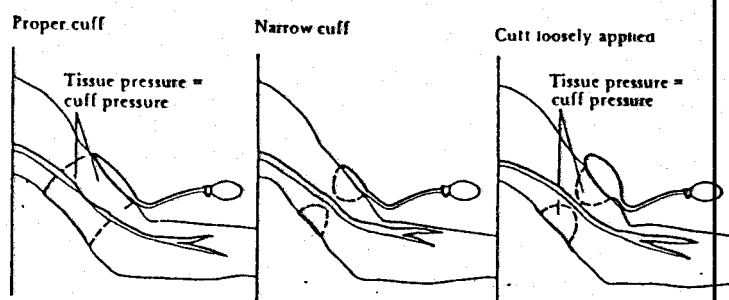
II.6. SISTEM PNEUMATIK

Untuk pengukuran tekanan darah dengan metoda tak langsung, diperlukan sistem pneumatik yang bekerja sama dengan transducer tekanan dan transducer bunyi Korotkoff, bagian pneumatis tersebut berupa cuff, katup, pompa / kompresor, penahan aliran udara. Untuk menghubungkan antara bagian pneumatik digunakan selang

II.6.1. Cuff

Cuff digunakan untuk melakukan penekanan terhadap arteri (biasanya brachialis artery, yaitu arteri yang ada pada lengan kanan / kiri). Tekanan yang dikenakan pada dinding artery diasumsikan sama dengan tekanan cuff¹⁹, walaupun kenyataannya tekanan cuff disalurkan melalui jaringan tubuh diantara cuff dengan arteri. Cuff terdiri dari kantong udara dan kain tempat kantong udara yang dibalutkan pada lengan. Gambar 2.35 menunjukkan pengaruh dari penggunaan cuff yang tepat dan tak tepat. Penggunaan sistem cuff yang tepat akan meneruskan tekanan cuff langsung pada jaringan tubuh yang melingkupi arteri, sehingga tekanan jaringan tersebut terhadap arteri diharapkan sama dengan tekanan cuff. Penggunaan cuff yang sempit atau terlalu longgar akan menyebabkan gangguan penyaluran tekanan cuff terhadap jaringan disekitar arteri. Pada umumnya dapat diterima bahwa lebar cuff sekitar 0.4 kali keliling cuff yang terpanjang. Cuff hendaknya ditempatkan sejajar dengan jantung untuk menghindari efek hidrostatik.

¹⁹ John G. Webster, op.cit, hal. 374

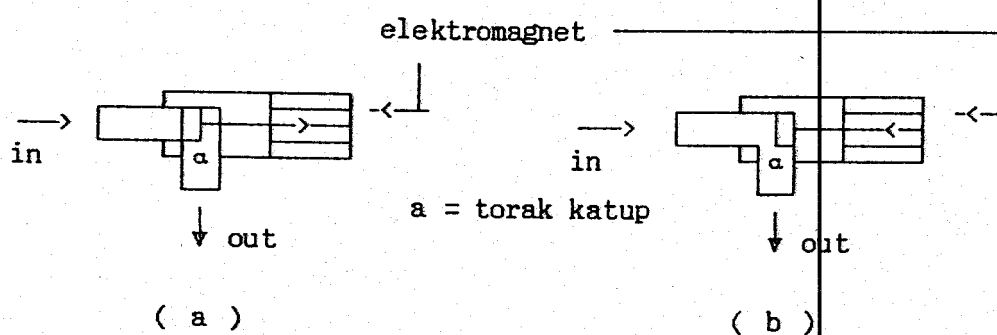


Gambar 2.35 Efek Dari Penggunaan Cuff

II.6.2. Katup Udara

Untuk mengontrol aliran udara maka digunakan katup udara yang digerakkan secara listrik. Pada dasarnya ada dua jenis katup, yaitu:

1. Katup dalam keadaan normal terbuka (normally open), katup ini akan terbuka pada saat tak ada catu daya dan tertutup pada saat ada catu daya.
2. Katup dalam keadaan normal tertutup (normally close). katup ini



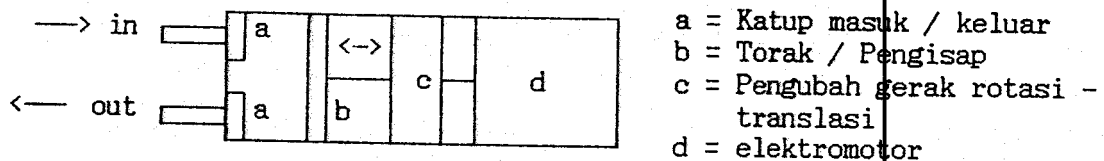
Gambar 2.36 (a) Katup Normally Close (b) Katup Normally Open
akan tertutup jika tak ada catu daya dan akan terbuka saat ada catu daya.

Secara blok diagram katup dapat digambarkan pada gambar 2.36

Arah panah dalam elektromagnet menyatakan arah gerakan jika elektromagnet diberikan catu daya, jika catu daya diputuskan maka pegas akan mengembalikan torak katup pada posisi normal.

II.6.3. Pompa Udara / Kompresor

Untuk menyediakan tekanan udara bagi cuff maka digunakan kompresor yang akan memompa udara dari luar dan dimampatkan pada cuff atau reservoir sebelum dialirkan pada cuff. Secara blok diagram kompresor / pompa udara dapat digambarkan seperti gambar 2.37.

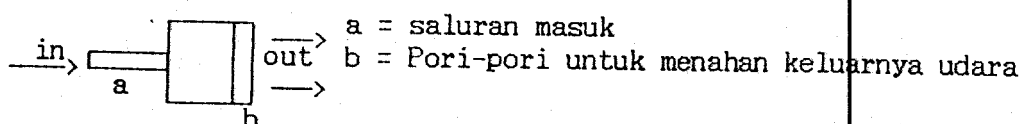


Gambar 2.37 Pompa / Kompresor Udara

Pada saat motor berputar (putaran motor tetap kesatu arah tertentu), oleh susunan mekanik c maka gerak putar ini diubah jadi gerak translasi dan menggerakkan torak b ke kiri dan ke kanan, pada saat torak b bergerak ke kanan maka katup masuk terbuka dan udara masuk ke selinder sementara itu katup keluar tertutup, pada saat torak b bergerak ke kiri maka katup masuk tertutup udara terdesak keluar selinder lewat katup keluar yang pada saat itu terbuka.

II.6.4. Penahan Aliran Udara

Pada metoda pengukuran tekanan darah dengan cara auskultasi maka udara dikeluarkan secara pelan-pelan dari cuff (penurunan tekanan 2 - 3 mmHg/detik). Untuk maksud tersebut udara dikeluarkan lewat penahan aliran udara yang secar blok diperlihatkan pada gambar 2.38 .



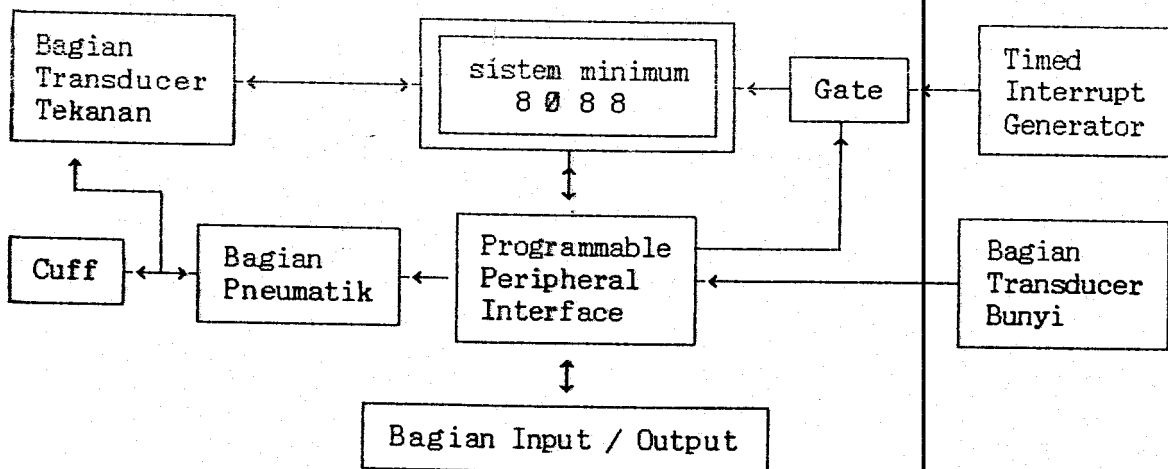
Gambar 2.38 Penahan Aliran Udara

BAB III

PERENCANAAN DAN PEMBUATAN HARDWARE

III.1. PENGUKUR TEKANAN DARAH OTOMATIS, PERIODIK, DAN KONTINU

Blok diagram sistem pengukur tekanan darah diperlihatkan pada gambar 3.1. Tekanan cuff disensor lewat transducer tekanan yang merupakan elemen sensor kapasitansi, sedangkan bunyi korotkoff disensor dengan kristal mikropon yang merupakan elemen piezoelektrik. Sebagai basis waktu adalah timer yang setiap 2 ms melakukan interrupt ke NMI. Untuk pengaturan tekanan cuff diadakan sistem pneumatik yang terdiri dari katup, pompa udara, dan penahan aliran udara. Masukkan dan keluaran informasi antara minimum sistem dan operator dilakukan lewat bagian input / output. Untuk menghubungkan peralatan peripheral dan sistem minimum digunakan PPI 8255.

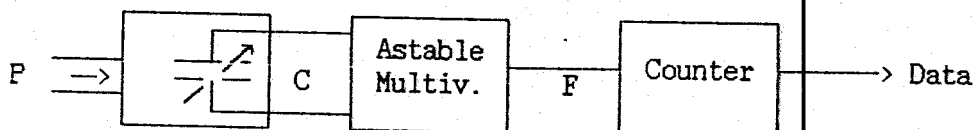


Gambar 3.1 Blok Diagram Sistem Pengukur Tekanan Darah

III.1.1. Bagian Transducer Tekanan

Bagian ini berfungsi untuk mengukur besar tekanan udara yang ada

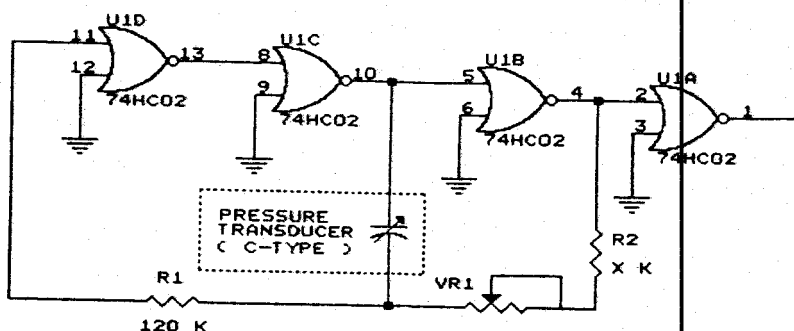
di dalam cuff pada saat tertentu. Transducer yang dipakai merupakan transducer kapasitansi. Besarnya kapasitansi transducer ketika tekanan relatif yang masuk 0 mmHg yaitu 18 pF, sedangkan jika tekanan relatif 300 mmHg kapasitansi transducer menjadi 36 pF. Kapasitansi transducer ini digunakan sebagai salah satu faktor penentu frekuensi dalam rangkaian astable multivibrator, sehingga besarnya frekuensi astable dapat berubah sesuai dengan besarnya kapasitansi tadi. Frekuensi dari astable multivibrator dimasukkan ke suatu counter sehingga banyaknya cacahan counter dalam selang waktu tertentu berubah sesuai dengan perubahan kapasitansi transducer (kapasitansi transducer berubah sesuai dengan tekanan). Software akan mengkonversikan jumlah cacahan counter pada tekanan tertentu menjadi harga tekanan relatif. Secara blok diagram bagian transducer tekanan ini diperlihatkan pada gambar 3.2.



Gambar 3.2 Bagian Transducer Tekanan

II.1.1.1. Astable Multivibrator

Bagian transducer tekanan ini dirancang untuk mengukur tekanan sampai 300 mmHg. Frekuensi output astable direncanakan berubah sebesar 1 KHz untuk setiap perubahan tekanan 1 mmHg. Rangkaian astable multivibrator ditunjukkan pada gambar 3.3. Harga R dihitung sesuai dengan rumus pada teori penunjang. Data untuk 74HC02 sbb:



Gambar 3.3 Rangkaian Astable Multivibrator Untuk Transducer Tekanan

$t_{pLH} = t_{pHL} = 8 \text{ ns}$, $V_{iH} = 3.15\text{V}$, $V_{iL} = 0.9\text{V}$, $V_{oH} = 4.4\text{V}$, $V_{oL} = 0.1\text{V}$

Jika harga-harga tersebut disubstitusikan pada rumus maka didapat, perioda, $T = RC \ln 57.33 + 48 \text{ ns}$. Frekuensi transducer $F = 1 / T$.

Kapasitansi transducer pada tekanan $0 \text{ mmHg} = 18 \text{ pF}$, pada tekanan $300 \text{ mmHg} = 36 \text{ pF}$. Perubahan frekuensi sebesar 1 KHz untuk perubahan tekanan 1 mmHg , jika F_1 adalah frekuensi pada 0 mmHg dan F_2 adalah frekuensi pada 300 mmHg , jadi :

$$\Delta F = 300 \text{ KHz}, \text{ atau } 1/T_1 - 1/T_2 = 300 \text{ KHz}$$

$$1 / (RC_1 \ln 57.33 + 48 \text{ ns}) - 1 / (RC_2 \ln 57.33 + 48 \text{ ns}) = 300000$$

untuk $C_1 = 18 \text{ pF}$, $C_2 = 36 \text{ pF}$, maka penyelesaian persamaan menghasilkan dua harga R yaitu

$$R = 21871,15692 \Omega \text{ maka } F_1 = 609 \text{ KHz}, F_2 = 309 \text{ KHz}$$

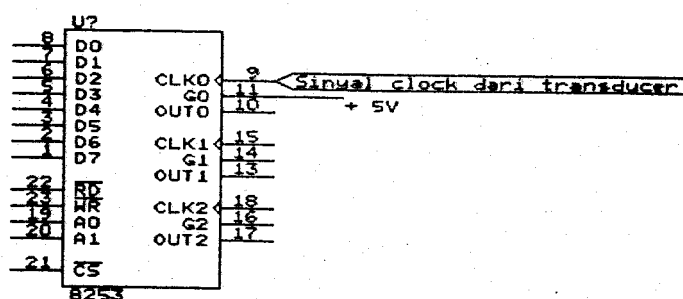
$$R = 9,916945012 \Omega \text{ maka } F_1 = 20,5 \text{ MHz}, F_2 = 20,2 \text{ MHz}$$

Pada pembuatan dipilih harga $R = 21871,15692 \Omega$ agar frekuensi tidak terlalu tinggi dan tidak diperlukan komponen yang khusus.

Untuk ini R_2 dipilih $20 \text{ K}\Omega$ dan $VR_1 = 3 \text{ K}\Omega$ digunakan untuk kalibrasi.

III.1.1.2. Counter

Counter direalisasi dengan chip 8253 counter 0 yang dioperasikan pada mode 0 yang dapat digunakan untuk mencacah turun. Frekuensi sinyal dari transducer 300 - 600 KHz maka untuk interval waktu 10 mS maka banyaknya cacahan 3000 - 6000. Untuk itu harga counter diberi nilai awal 9999 dan setelah 10 mS isi counter diambil, sehingga banyaknya cacahan selama itu sama dengan pengurangan isi counter.

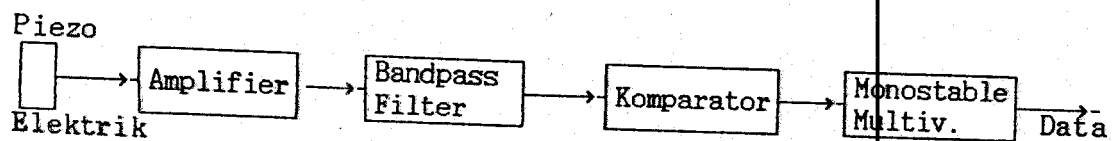


Gambar 3.4 Counter Untuk Bagian Transducer Tekanan

III.1.2. Bagian transducer Bunyi

Bagian ini digunakan untuk mendeteksi bunyi Korotkoff, bagian ini terdiri atas transducer piezoelektrik, non inverting amplifier, filter, komparator, dan monostable multivibrator. Sinyal dari piezoelektrik dikuatkan oleh amplifier. Bunyi Korotkoff berkisar antara 20 - 300 Hz sehingga untuk menghilangkan gangguan yang berfrekuensi diluar range frekuensi tersebut, sinyal dari amplifier dilewatkan filter bandpass 20 - 300 Hz. Komparator digunakan untuk mendeteksi kenaikan tegangan karena adanya bunyi korotkoff, dan mengabaikan kenaikan tegangan yang kecil karena noise (dalam hal ini tegangan referensi komparator diset lebih besar dari tegangan masuk yang disebabkan noise). Monostable ditrigger oleh sinyal dari

komparator, karena tegangan yang masuk komparator yang disebabkan bunyi korotkoff bersifat AC maka akan terjadi beberapa kali trigger dari komparator, untuk menyesuaikan hal ini monostable diset pada mode retriggerable, sinyal output monostable ini akan tetap high selama satu peristiwa bunyi korotkoff. Frekuensi terendah dari bunyi korotkoff sesuai dengan batas bawah frekuensi cutoff filter 20 Hz atau mempunyai periode 50 ms, untuk itu monostable harus mempertahankan keadaan setelah trigger selama 50 ms. Gambar 3.5 memperlihatkan bagian transducer bunyi.



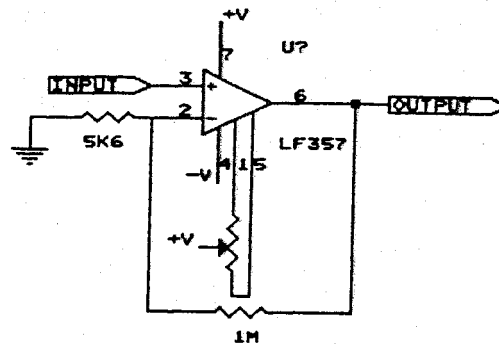
Gambar 3.5 Bagian Transducer Bunyi

III.1.2.1. Amplifier

Amplifier yang digunakan adalah non inverting amplifier yang direalisasi dengan Op-amp LF 357. Digunakan non inverting amplifier dan Op-amp dari jenis FET sebab impedansi output piezo elektrik tinggi sehingga harus digunakan amplifier yang berimpedansi tinggi juga. Dari teori penunjang diketahui tegangan output berorde mV, untuk itu dipilih penguatan amplifier sekitar 180, sehingga output amplifier akan berorde ratusan mV (cukup untuk pengolahan sinyal lebih lanjut) Untuk rangkaian seperti pada gambar 3.6, maka penguatan amplifier,

$$A = 1 \text{ M} / 5,6 \text{ K} + 1$$

$$= 179.5$$



Gambar 3.6 Rangkaian Amplifier

III.1.2.2. Filter

Untuk meredam noise yang ada digunakan band pass filter. Sesuai teori metoda auskultasi maka deteksi bunyi harus baik pada frekuensi 20 - 300 Hz, maka dirancang filter bandpass dengan frekuensi cutoff 20 dan 300 Hz. Filter terdiri atas Low pass filter disusul high pass filter dari jenis Butterworth 40 dB / dec.

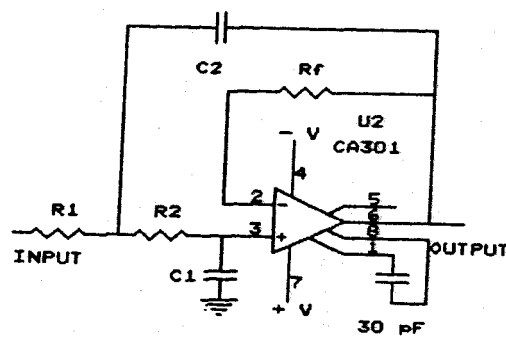
Gambar 3.7 menunjukkan low pass filter, penentuan harga komponennya adalah:

1. Frekuensi cutoff 300 Hz, jadi $\omega_c = 1885 \text{ rad / det}$
2. R1 dan R2 dipilih sama dengan nilai R, yaitu 27 K
 $R_f = 2R = 54 \text{ K}$
3. $C1 = 0,707 / \omega_c.R = 0,707 / (1885 * 27K) = 14 \text{ nF}$
4. $C2 = 2C1 = 28 \text{ nF}$

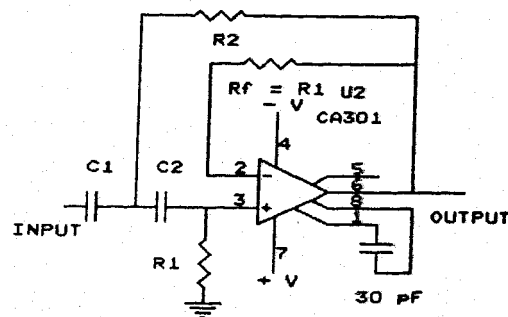
Gambar 3.8 menunjukkan high pass filter, penentuan harga komponennya adalah:

1. Frekuensi cutoff 20 Hz, jadi $\omega_c = 125,66 \text{ rad / det}$
2. C1 dan C2 dipilih sama dengan nilai C, yaitu 0,1 uF

3. $R_1 = 1,414 / \omega_c \cdot C = 1,414 / (125,66 \cdot 0,1 \mu F) = 112,525 \text{ K}$
4. $R_2 = 0,5 R_1 = 56 \text{ K}$
5. $R_f = R_1 = 112,525 \text{ K}$



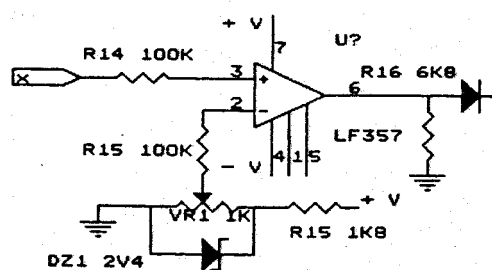
Gambar 3.7 Filter Low Pass -40 db / dec



Gambar 3.8 Filter High Pass 40 db / dec

III.1.2.3. Komparator

Komparator digunakan untuk menindas sinyal noise yang kecil (dibawah level tegangan tertentu), disamping itu komparator digunakan untuk mentrigger monostable multivibrator. Gambar 3.9 menunjukkan komparator yang digunakan, level tegangan referensinya dihasilkan rangkaian resistor dan diode zener. Diode zener yang dipakai bernilai 2,4 V dengan daya 0,5 Watt untuk membatasi agar daya yang masuk lebih rendah dari 0.5 Watt digunakan resistor 1,8 K, jadi daya pada diode, $P_z = 2,4 * ((12-2,4) / 1,8 K) = 0.0128$ Watt. Tegangan referensi komparator dapat diatur oleh variabel resistor 1 K. Karena tegangan untuk mentrigger monostable multivibrator adalah level positif saja maka digunakan diode untuk mendeagah tegangan negatif masuk ke monostable multivibrator.

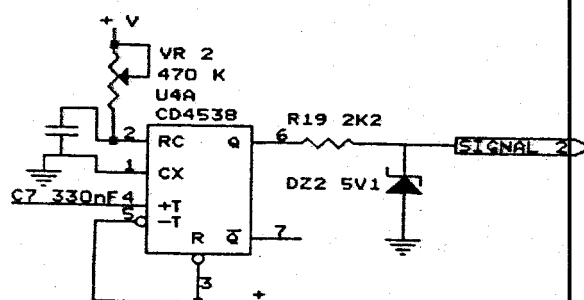


Gambar 3.9 Rangkaian Komparator

III.1.2.4. Monostable Multivibrator

Monostable multivibrator digunakan untuk menahan level tegangan digital keluaran komparator selama waktu tertentu sehingga level tegangan ini dapat dibaca sebagai data yang stabil. Jika ada pulsa bunyi korotkoff maka keluaran komparator akan berayun positif - negatif dan sebaliknya karena bunyi korotkoff tersebut bersifat AC.

Monostable multivibrator yang digunakan adalah IC 4538 (CMOS), yang digunakan pada mode trigger positif dan bersifat dapat ditrigger ulang. Lama waktu menahan level tegangannya, $T = RC$. Dipilih nilai C sebesar 330 nF dan R berupa variabel sebesar 470k, maka T maksimum adalah 155 mS. Untuk penyesuaian level tegangan data maka tegangan outputnya dibatasi maksimum 5,1 volt.



Gambar 3.10 Rangkaian Monostable Multivibrator

III.1.3. Sistem Minimum 8088

Sistem ini terdiri atas pembangkit clock 8284 yang membangkitkan frekuensi clock 4,773 MHz (hasil dari frekuensi kristal 14,31818 MHz dibagi tiga), mikroprosesor 8088 yang dikerjakan pada mode minimum, address latch, data transceiver, decoder untuk mengakses memori dan IO, memori dan decodernya (memori terdiri atas EPROM berkapasitas 8 K dan RAM yang berkapasitas 2 x 2 K. Sistem ini menerima interrupt dari Timed Interrupt Generator setiap 2 ms, interrupt ini dimasukkan ke NMI, dan dijadikan sebagai basis waktu untuk operasi sistem minimum.

III.1.4. Timed Interrupt Generator

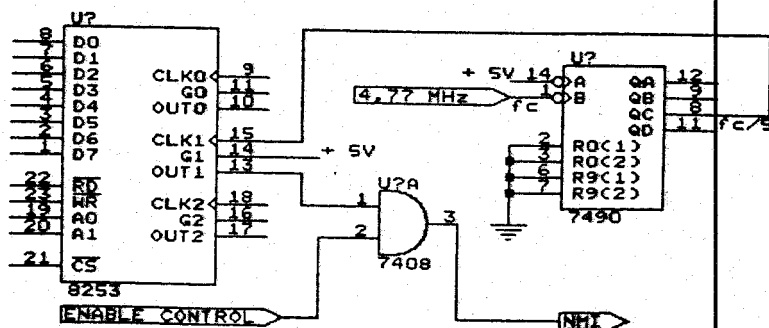
Untuk menyediakan basis waktu yang tepat dan tetap bagi sistem

minimum maka direncanakan timed interrupt generator yang akan menghasilkan sinyal interrupt dengan periode 2 ms, hal ini dianggap memadai, jika dipakai sebagai standar waktu untuk menghitung heart rate sampai 300 kali / menit, yang berarti berfrekuensi 5 Hz, atau mempunyai periode 200 ms.

Kesalahan yang timbul sebesar : $2 / 200 * 100 \% = 1 \%$

Bagian ini direalisasi dengan IC Programmable Interval Timer (PIT) 8253 yang dikerjakan pada mode 2, yaitu sebagai Timed Interrupt Generator. Karena input clock 8253 maksimum hanya 2,6 MHz, sedangkan sinyal clock didapat dari 8284 yang berfrekuensi 4,772726667 MHz maka sebelum masuk ke 8253 sinyal clock dibagi 5 oleh counter 74LS90. Jadi sinyal clock yang masuk ke 8253 berfrekuensi 0,9545453334 MHz. Supaya output 8253 dapat menghasilkan sinyal interrupt setiap 2 ms (frekuensi 500 Hz) maka isi awal counter (yang merupakan pembagi frekuensi clock yang masuk) $954545,3334 / 500 = 1909$.

Untuk mencegah masuknya sinyal interrupt ke NMI pin dari CPU sebelum inisialisasi (vektor interrupt dan subroutine untuk melayani NMI), maka sinyal interrupt dienable oleh gate (gate merupakan gerbang AND) setelah proses inisialisasi selesai. Basis waktu juga digunakan oleh



Gambar 3.11 Rangkaian Penghasil Sinyal Interrupt

software sehingga alat ini dapat berfungsi secara periodik dan kontinu untuk waktu tertentu. Penggunaan lain dari basis waktu yaitu untuk menentukan waktu dari counter dalam mencacah frekuensi yang masuk dari astable multivibrator pada bagian transducer tekanan.

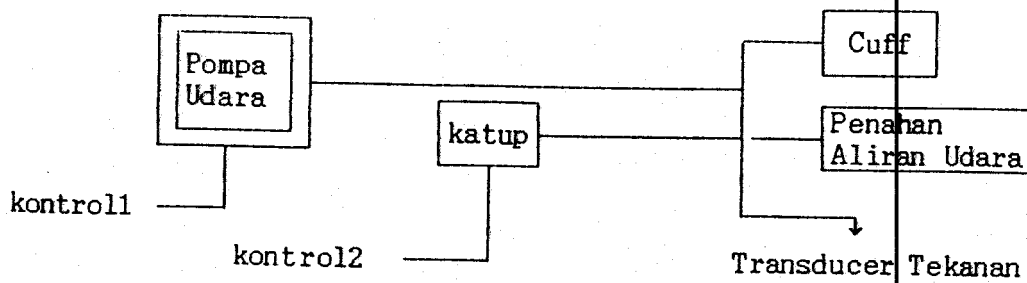
III.1.5. Programmable Peripheral Interface

Untuk mengatur keluar masuknya data antara sistem minimum dan beberapa peralatan / rangkaian peripheral maka digunakan PPI 8255 yang dikerjakan pada mode 0. PPI 8255 digunakan sebagai port untuk output / input tanpa sinyal handshaking. Untuk menangani bagian I/O maka digunakan 2 buah PPI 8255.

III.1.6. Bagian Pneumatik Dan Cuff

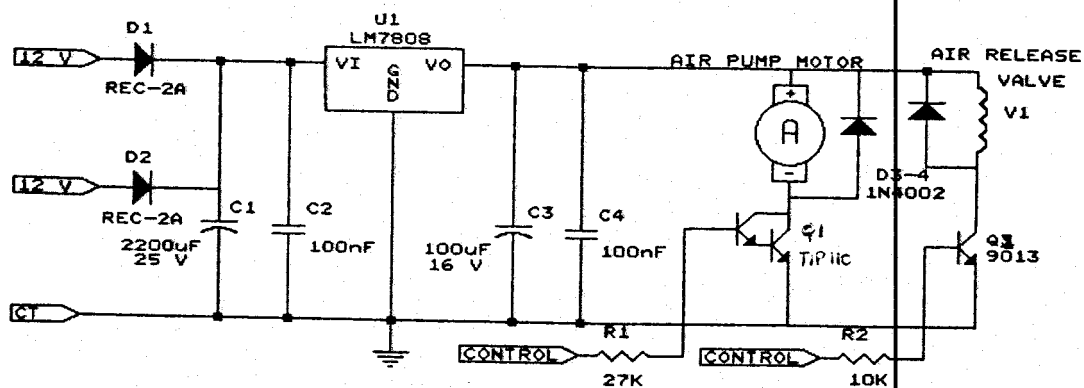
Dalam metoda determinasi tekanan darah secara Auskultasi diperlukan pengaturan tekanan cuff, untuk proses otomatis maka diperlukan pompa udara / kompressor, katup, penahan aliran udara. Pada awal proses determinasi cuff diberi tekanan udara dengan cara menghidupkan pompa, jika tekanan sudah mencapai harga yang dikehendaki (data tekanan awal dimasukkan lewat keyboard). Untuk proses determinasi selanjutnya tekanan cuff ditentukan sebesar 30 mmHg diatas tekanan sistolik yang terdeteksi sebelumnya.

Penurunan tekanan cuff sebesar 2 - 3 mmHg / s dilakukan dengan mengeluarkan udara lewat penahan aliran udara. Jika proses determinasi telah selesai maka sisa udara di dalam cuff langsung dikeluarkan melalui katup.



Gambar 3.12 Bagian Pneumatik Dan Cuff

Sinyal kontrol didapat dari PP1 8255, gambar 3.12. menunjukkan rangkaian untuk mengendalikan pompa dan katup, beserta catu daya yang digunakan.



Gambar 3.13 Rangkaian Pompa Udara dan Katup

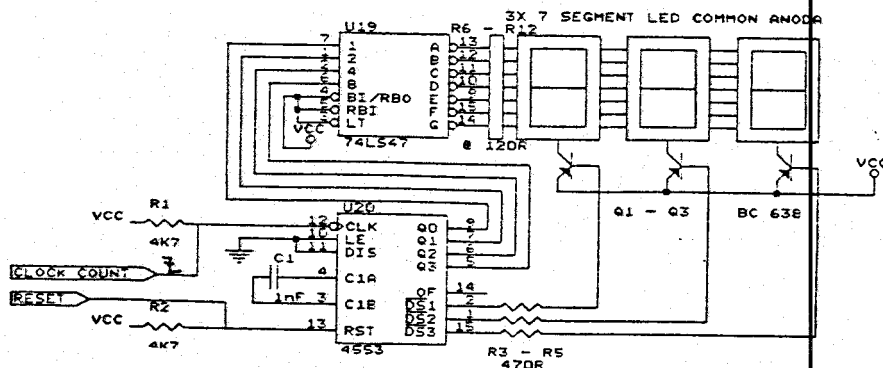
III.1.7. Bagian Input / Output

Bagian ini merupakan perantara masukan atau keluaran informasi / data antara minimum sistem dan operator. Agar komunikasi lebih efisien maka sistem akan mengemukakan data yang diperlukan lewat display dan operator menjawabnya lewat keyboard. Speaker digunakan untuk memberi peringatan tentang sesuatu.

Bagian input terdiri atas keyboard yang terdiri atas 16 switch spst, proses scanning keyboard dijalankan oleh CPU sehingga tombol yang telah ditekan operator dapat diketahui sistem. Penggunaan tombolnya sebagai berikut:

1. Sepuluh tombol digunakan untuk angka 0 - 9.
2. Dua tombol untuk menjawab pertanyaan (ya / tidak, pria / wanita, dsb).
3. Satu tombol untuk memasukkan data yang sudah benar.
4. Satu tombol untuk mengulangi jenis data yang membutuhkan jawaban.
5. Satu tombol untuk memulai alat beroperasi.
6. Satu tombol dicadangkan.

Bagian output terdiri atas display numerik (terdiri atas 4 buah yang masing-masing terdiri atas tiga digit), display ini digunakan untuk menampilkan harga-harga sistolik, diastolik, mean artery pressure, dan heart rate. Masing - masing unit display ini terdiri atas counter dan decoder BCD ke 7 segment, data dikirim ke unit display lewat masukan clock dari counter. Counter CMOS 4553 dilengkapi dengan internal multiplex dan scanning kontrol sehingga isinya langsung dapat ditampilkan ke seven segmen setelah keluaran

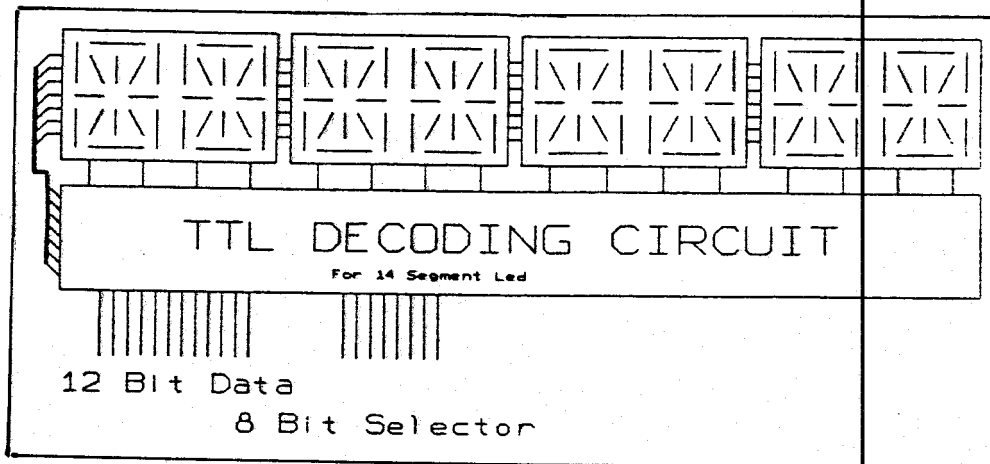


Gambar 3.14 Rangkaian Tampilan Tiga Digit

data BCD nya didecoder ke seven segment oleh 74LS47. Sinyal reset dan clock count didapat dari PPI 8255.

Display alfa numerik terdiri atas 8 buah karakter yang akan mengemukakan jenis data yang memerlukan jawaban operator, pada akhir proses determinasi display ini akan menyarankan hasil diagnosa dari pasien. Display ini dijalankan dengan proses scanning dan sedikit decoding sebelum sinyal dari PPI digunakan untuk menjalankan alfa numerik LED.

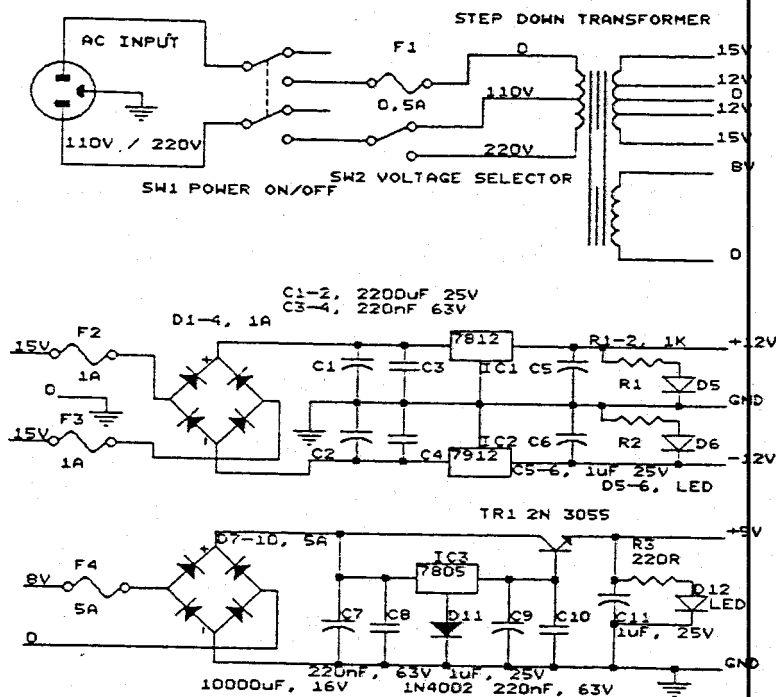
Tabel 3.1. Tampilan Alfa Numerik



Speaker sebagai sumber alarm / peringatan misalnya pada saat alat mulai bekerja, jika ada denyut nadi yang terdeteksi selama proses determinasi, saat alat mengakiri proses determinasi dan hasil ditampilkan, jika terjadi ketidaknormalan hasil pengukuran. Frekuensi yang digunakan untuk membunyikan speaker didapat dari PIT 8253 yang dioperasikan pada mode tiga, yaitu Square Wave. Untuk membunyikan speaker sinyal dari 8253 dibuffer dengan transistor NPN 9013.

III.1.8. Catu Daya

Karena hampir semua rangkaian terpadu yang digunakan mempunyai level tegangan TTL dan ada beberapa operasional amplifier, maka catu daya direncanakan menyediakan tiga level tegangan yaitu + 12V, + 5V, dan - 12V. Besar arus maksimum untuk + / - 12V yaitu 1A, sedangkan untuk +5V arus maksimumnya 5A. Khusus untuk catu daya pompa dan katup dibuatkan secara terpisah sebesar + 8V

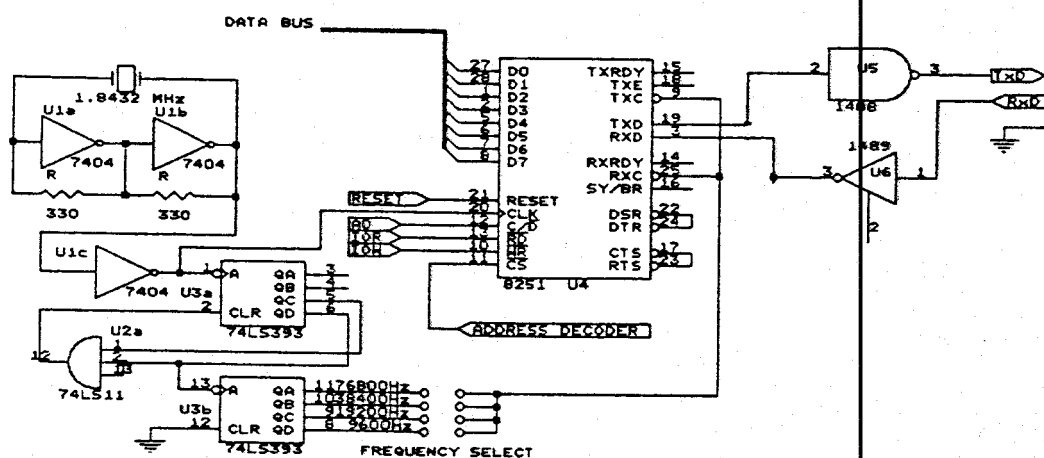


Gambar 3.15 Rangkaian Catu Daya

III.2. KOMUNIKASI SERIAL DENGAN IBM PC

Fasilitas yang disediakan untuk pengolahan data lebih lanjut seperti tampilan grafik dilayar, data yang disimpan sebagai file, dan sebagainya akan lebih mudah dilakukan oleh personal komputer. Karena itu maka diadakan komunikasi secara serial dengan personal komputer

Transfer data secara serial menggunakan fasilitas yang sudah ada pada PC yaitu komunikasi serial dengan RS-232. Pada minimum sistem digunakan rangkaian terpadu USART 8251, pengubah level tegangan TTL ke RS-232 (MC 1488), dan pengubah level tegangan RS-232 ke TTL (MC 1489).



Gambar 3.16 Rangkaian Komunikasi Serial RS 232

BAB IV

PERENCANAAN DAN PEMBUATAN SOFTWARE

IV.1. SOFTWARE MINIMUM SISTEM 8088

Software yang disimpan di-Eprom pada minimum sistem 8088 berfungsi untuk mengendalikan proses kerja hardware sesuai dengan langkah-langkah yang diperlukan untuk operasi alat. Flowchart pada gambar 4.1 menunjukkan langkah-langkah yang diperlukan untuk mengoperasikan bagian hardware. Langkah-langkah ini diwujudkan seperti pada program utama. Untuk mempermudah pemrograman maka program utama terdiri atas prosedur-prosedur untuk menjalankan tugas tertentu. Beberapa prosedur dimungkinkan memanggil prosedur lain.

Eprom juga berisi data medis range tekanan normal pada usia dan jenis kelamin tertentu, data ini berguna mengambil kesimpulan keadaan pasien setelah diketahui tekanan darahnya.

IV.1.1. Prosedur Inisialisasi Hardware

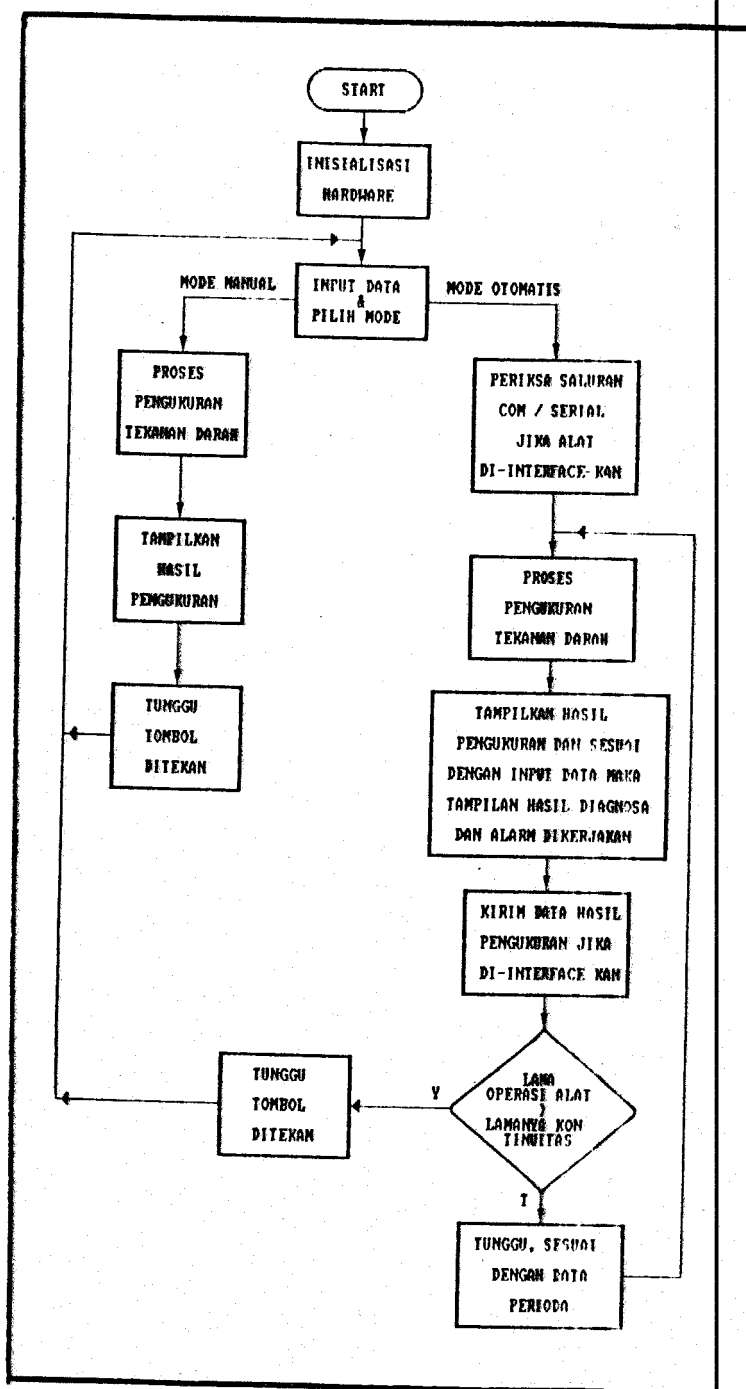
Prosedur ini berfungsi mempersiapkan kerja hardware, seperti memesan lokasi memori, menentukan segment memori, menentukan mode operasi PPI 8255, PIT 8253, USART 8251, mempersiapkan interrupt ke NMI, mereset display numerik.

Prosedur ini ada pada awal program dan hanya sekali dijalankan.

IV.1.2. Prosedur Masukkan Data

Prosedur ini berguna untuk memasukkan data, seperti data mode

kerja alat (mode otomatis atau manual), data tekanan awal cuff,



Gambar 4.1 Flowchart Software Eprom

data umur dan jenis kelamin pasien, data perioda kerja untuk melakukan pengukuran tekanan darah (data bersatuan menit), data kontinuitas

kerja alat (data bersatuan jam), data untuk membunyikan alarm jika hasil diagnosa tidak normal, data untuk menampilkan hasil diagnosa setelah pengukuran, data untuk melakukan komunikasi dengan komputer, data untuk mengetes saluran RS 232.

Nama prosedur ini : data_input

IV.1.3. Prosedur Tampilan Kata

Prosedur ini berguna untuk menampilkan kata-kata pada alfa numerik display, prosedur melakukan scanning display dan memeriksa penekanan tombol (dalam hal ini prosedur ini memanggil prosedur scanning keyboard)

Nama prosedur ini : word_expose

IV.1.4. Prosedur Scanning Keyboard

Prosedur ini melakukan scanning pada 4 x 4 matrix keyboard, hasil scanning adalah data ada tidaknya tombol yang ditekan, dan jenis tombol yang ditekan tersebut.

Nama prosedur ini : keyb_scan

IV.1.5. Prosedur Penundaan Waktu

Prosedur ini berguna untuk melakukan penundaan waktu pada suatu proses, misalnya penundaan waktu untuk memberi kesempatan alfa numerik display menyala.

Nama prosedur ini : ndelay, fdelay

IV.1.6. Prosedur Tampilan Keadaan Alat

Prosedur ini berguna untuk mengetahui keadaan operasi alat misalnya alat dalam keadaan menunggu untuk periode pengukuran berikutnya, alat sedang bekerja pada mode otomatis atau manual.

Nama prosedur ini : waitexpose, autoexpose, manualexpose

IV.1.7. Prosedur Bunyi Speaker

Prosedur ini berguna untuk membunyikan speaker dengan variasi bunyi yang bermacam-macam, seperti bunyi yang menandakan keyboard tertekan, bunyi untuk menandakan terdeteksinya pulsa tekanan / denyut nadi, bunyi yang menandakan kegagalan alat dalam pengukuran tekanan, bunyi awal alat bekerja, bunyi akhir operasi alat, dsb

Nama prosedur ini : start_sound, end_sound, pulse_sound, keyb_sound, fail_sound

IV.1.8. Prosedur Komunikasi Serial RS 232

Prosedur ini terdiri atas prosedur pengecekan saluran RS 232, prosedur sinkronisasi pada awal pengiriman data, prosedur pengiriman data, dan prosedur penerimaan data.

Nama prosedur ini : usart_out, usart_in, synchron, com_test

IV.1.9. Prosedur Pengukuran Tekanan Darah

Prosedur ini berguna untuk mengendalikan hardware dalam melakukan pengukuran tekanan darah, prosedur ini memanggil prosedur lainnya seperti prosedur untuk menentukan tekanan cuff, prosedur bunyi yang menandakan proses pengukuran gagal, prosedur untuk mengetahui frekuensi dari bagian transducer tekanan pada saat tertentu, prosedur

untuk mengubah harga frekuensi transducer tekanan ke harga tekanan yang dideteksi transducer tekanan tersebut, prosedur untuk menghasilkan bunyi yang menandakan adanya pulsa tekanan yang terdeteksi, prosedur untuk menampilkan harga tekanan cuff ke numerik display, prosedur untuk membuka katup setelah proses pengukuran selesai sehingga tekanan cuff langsung hilang. Adapun langkah-langkah yang dilakukan prosedur ini adalah, sebagai berikut:

1. Memompa cuff sampai pada tekanan tertentu sesuai dengan data yang tersedia.
2. Menunggu sampai pulsa tekanan terdeteksi, waktu untuk menunggu ini dibatasi 15 detik, waktu ini adalah waktu tunggu penurunan tekanan cuff sampai pada tekanan sistolik. Waktu 15 detik ini ditentukan untuk memberi kesempatan tekanan cuff turun sampai sekitar 45 mmHg sehingga pulsa sistolik dapat terdeteksi (pada keadaan normal tekanan awal cuff 30 mmHg diatas tekanan sistolik dan penurunan tekanan cuff 3 mmHg/detik).
3. Tekanan cuff pada saat pulsa pertama terdeteksi ditetapkan sebagai tekanan sistolik.
4. Menentukan waktu antara dua pulsa tekanan yang terdeteksi, waktu tersebut di rata-rata untuk menentukan heart rate.
5. Jika waktu antara dua pulsa tekanan lebih besar dari 2 detik maka tekanan pada pulsa terakhir dianggap tekanan diastolik. Harga 2 detik ditentukan agar alat dapat mengukur heart rate paling rendah dengan perioda 2 detik atau 30 beat/min.
6. Jika proses pengukuran gagal, misalnya waktu tunggu pulsa tekanan lebih besar dari 15 detik (pulsa tekanan tak terdeteksi dalam

waktu 15 detik sejak tekanan cuff ditetapkan) atau banyaknya pulsa yang terdeteksi kurang dari 4 sehingga pengukuran tidak presisi maka proses pengukuran tekanan diulang dari awal sebanyak 2 kali. Tiap proses pengukuran gagal bunyi speaker yang menandakan kegagalan dijalankan. Jika setelah diusahakan sebanyak 2 kali namun tetap gagal maka alat akan membunyikan speaker yang menandakan kegagalan dalam waktu yang agak lama, dan menampilkan tulisan pada alfa numerik display. Data yang ditampilkan / dikirim lewat RS 232 adalah data-data lama, jika kegagalan terjadi pada saat pertama sekali sehingga belum ada data maka semua data yang ditampilkan / dikirim diberi nilai 1.

7. Pada akhir proses pengukuran maka ditampilkan data-data hasil pengukuran ke numerik display, data tersebut adalah tekanan sistolik, diastolik, MAP, dan heart rate.

Nama prosedur ini : monitor

IV.1.10. Prosedur Penyiapan Data Untuk RS 232

Karena data yang didapat lebih besar dari 255 (8 bit) sedangkan pengiriman lewat RS 232 hanya berupa data 8 bit maka data 16 bit yang akan dikirim dipisah dulu menjadi 2 data 8 bit.

Nama prosedur ini : separate

IV.1.11. Prosedur Pengambilan Keputusan

Prosedur ini menghasilkan keputusan keadaan pasien berdasarkan data hasil pengukuran setelah dibandingkan dengan data keadaan normal menurut umur dan jenis kelamin. Jadi keputusan yang diambil terbatas

berdasarkan umur dan jenis kelamin saja.

Nama prosedur ini : decision

IV.1.12. Prosedur Tampilan Kalimat

Prosedur ini berguna untuk menampilkan kalimat, karena alfa numerik display terbatas 8 buah saja, maka kalimat ditampilkan kata per kata dari awal sampai akhir.

Nama prosedur ini : diagnose

IV.1.13. Prosedur Harga Tekanan Awal Cuff

Pada saat pertama kali alat melakukan proses pengukuran maka tekanan awal cuff didapat dari masukkan data yang prosesnya diatur oleh prosedur ini.

Nama prosedur ini : initialpress

IV.1.14. Prosedur Tekanan Cuff

Prosedur ini menjalankan pompa sampai tekanan cuff mencapai harga yang ditetapkan yaitu pada awal ditentukan oleh masukkan data sedangkan untuk pengukuran berikutnya pompa dijalankan sampai tekanan cuff 30 mmHg diatas tekanan sistolik.

Nama prosedur ini : setpressure

IV.1.15. Prosedur Tampilan Numerik

Prosedur ini berguna untuk menampilkan data numerik ke seven segmen display. Prosedur menghasilkan sinyal reset lalu clock yang banyaknya sama dengan harga bilangan yang akan ditampilkan. Untuk

mereset keempat grup numerik display digunakan prosedur reset (dreset)
 Nama prosedur ini : expose1,expose2,expose3,expose4,dreset

IV.1.16. Prosedur Frekuensi Transducer

Prosedur ini berguna untuk menghitung frekuensi bagian transducer tekanan pada tekanan tertentu. Hal ini dilakukan dengan cara menghitung banyaknya cacahan pulsa dari bagian transducer selama 10 ms (dalam hal ini prosedur mengontrol counter dari 8253). Hasil perhitungannya berupa banyaknya cacahan selama 10 ms.

Nama prosedur ini : get_data

IV.1.17. Prosedur Konversi Frekuensi ke Tekanan

Prosedur ini mengubah harga cacahan counter selama 10 ms ke harga tekanan. Pada uji coba transducer tekanan maka didapat data yang linear antara cacahan pada selang waktu tertentu (frekuensi) dengan harga tekanan yang diukur, data uji coba didapat dengan menaikkan tekanan dari 0 sampai 300 mmHg dan dicatat frekuensinya tiap kenaikan tekanan 10 mmHg. Karena data cenderung linear maka dapat dicari persamaan linear antara frekuensi dan tekanan. Pada perhitungan konversi ini digunakan cara yang lebih teliti yaitu interpolasi linear antara dua harga frekuensi yang didapat dari uji coba tadi. Jadi pertama frekuensi yang akan dikonversikan ditetapkan berada dalam range data tertentu dari data uji coba transducer, misalnya frekuensi x berada dalam range frekuensi (cacahan tiap 10 ms) 6607 dan 6490, yaitu untuk range tekanan 0 mmHg dan 10 mmHg, maka dibuatlah persamaan interpolasi (linear) antara dua titik tersebut dan setelah itu

frekuensi yang akan dikonversikan dimasukkan ke dalam persamaan tadi sehingga didapat harga tekanannya.

Nama prosedur ini : convert

IV.1.18. Prosedur NMI

Prosedur ini digunakan untuk melayani NMI, tiap ada NMI maka isi t_{low} dinaikkan satu, selang waktu NMI 2 mS jadi t_{low} mempunyai harga satuan waktu 2mS, untuk menyatakan waktu satu detik sebesar 500 satuan t_{low} , waktu satu menit sebesar 30000 t_{low} , setelah t_{low} berharga 30000 maka t_{low} direset dan isi t_{high} dinaikkan satu jadi t_{high} mempunyai satuan waktu 1 menit sehingga untuk waktu 24 jam dapat dinyatakan dengan 1440 t_{high} , setelah harga 1440 t_{high} direset lagi. tiap kali ada NMI flag F selalu diset (diberi nilai 1), hal ini berguna untuk awal penghitungan waktu yang harus menunggu NMI dulu.

Nama prosedur ini : NMI

IV.1.19. Prosedur Buka Katup

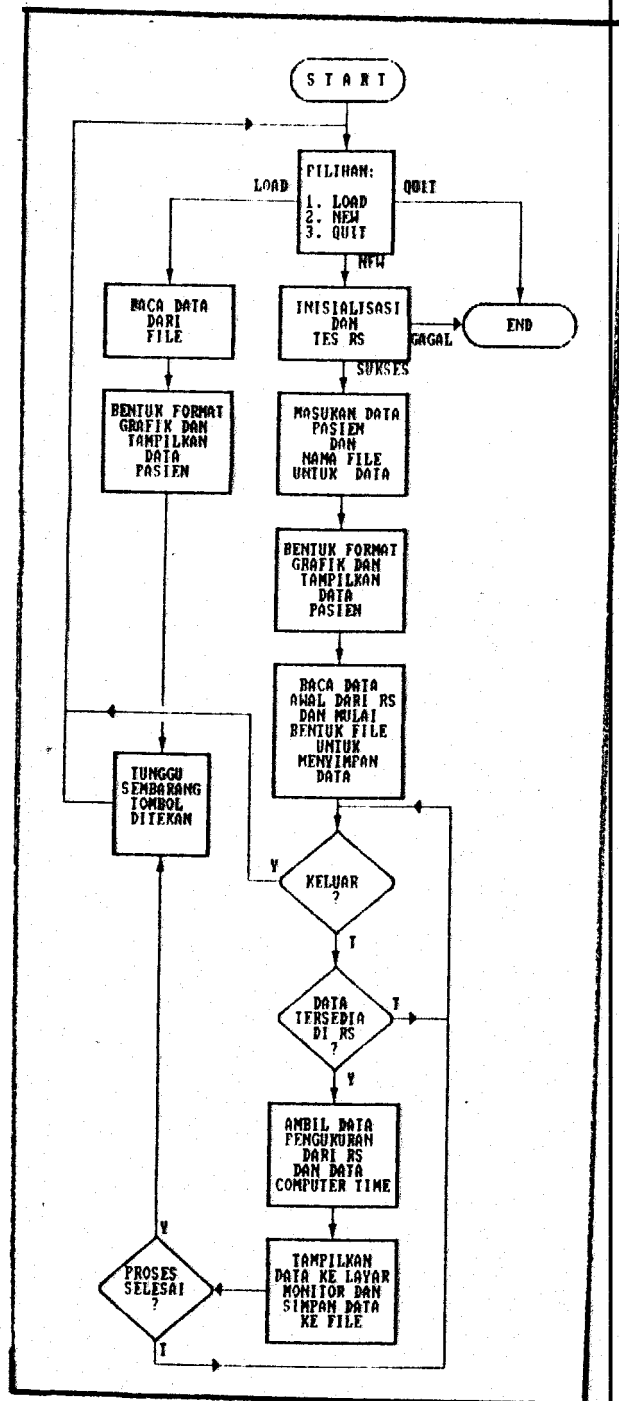
Prosedur ini berguna untuk membuka katup pada saat proses pengukuran selesai sehingga tekanan cuff langsung hilang.

Nama prosedur ini : openv

IV.2. SOFTWARE APLIKASI KE KOMPUTER PC

Pengolahan data lebih lanjut seperti pembuatan grafik dan penyimpanan ke file dapat dilakukan di komputer PC. Flowchart program ini dapat dilihat pada gambar 4.2. Program direalisasi dengan

bahasa Pascal.



Gambar 4.2 Flowchart aplikasi Komputer PC

BAB V

PENGUKURAN DAN UJI COBA ALAT

Pengukuran terutama dilakukan pada bagian transducer dan pengolah sinyalnya, dari hasil pengukuran diambil keputusan untuk menentukan hardware dan software yang cocok, sehingga pada uji coba alat dapat berfungsi dengan baik.

V.1. BAGIAN TRANSDUCER TEKANAN

V.1.1. Transducer Tekanan

Transducer tekanan yang digunakan merupakan elemen kapasitansi. Pengukuran transducer tekanan dilakukan dengan cara mengukur kapasitansinya pada saat besar tekanan tertentu diadakan. Hasil pengukurannya ditabelkan pada tabel 5.1.

Dari pengukuran diatas terlihat perubahan harga kapasitansi hanya sebesar 18 pF untuk perubahan tekanan 300 mmHg. Dari kenyataan diatas maka diputuskan kapasitansi transducer digunakan sebagai komponen kapasitansi pada astable multivibrator dengan frekuensi ribuan Hertz sehingga perubahan frekuensi terhadap perubahan tekanan nyata.

V.1.2. Output Frekuensi Astable Multivibrator

Dilakukan pengukuran output frekuensi astable multivibrator dimulai dari tekanan 0 mmHg sampai 300 mmHg, tiap harga tekanan dilakukan lima pengukuran lalu diambil harga rata-ratanya. Hasil pengukuran ditunjukkan pada tabel 5.2. Pengukuran dilakukan sebanyak lima kali, datanya ada pada lampiran.

Tabel 5.1. Data Pengukuran Kapasitansi Transducer Tekanan

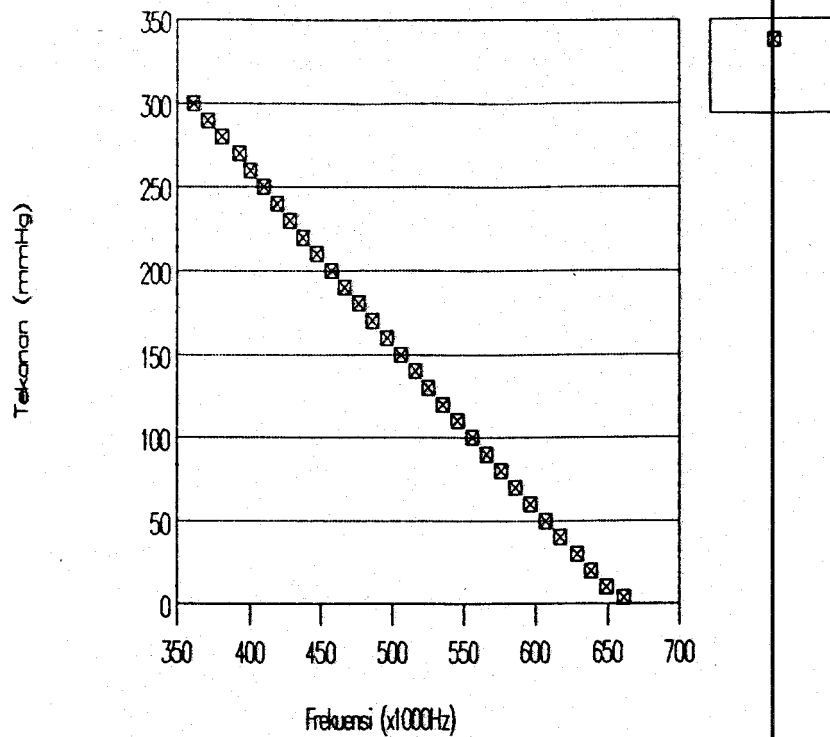
Tekanan (mmHg)	Kapasitansi (pF)	Tekanan (mmHg) C berubah
0	18	
10	18	
20	18	
30	18	
40	19	38
50	19	
60	19	
70	20	68
80	20	
90	20	
100	21	93
110	21	
120	22	117
130	22	
140	23	136
150	23	
160	24	157
170	24	
180	25	175
190	26	189
200	26	
210	27	205
220	28	219
230	29	229
240	29	
250	30	242
260	31	253
270	32	263
280	33	273
290	34	282
300	35	289
	36	298

Tester: Mastech Digital Instrument M 890 D, Resolusi 1 pF

Tabel 5.2. Data Pengukuran Frekuensi Output Astable Multivibrator

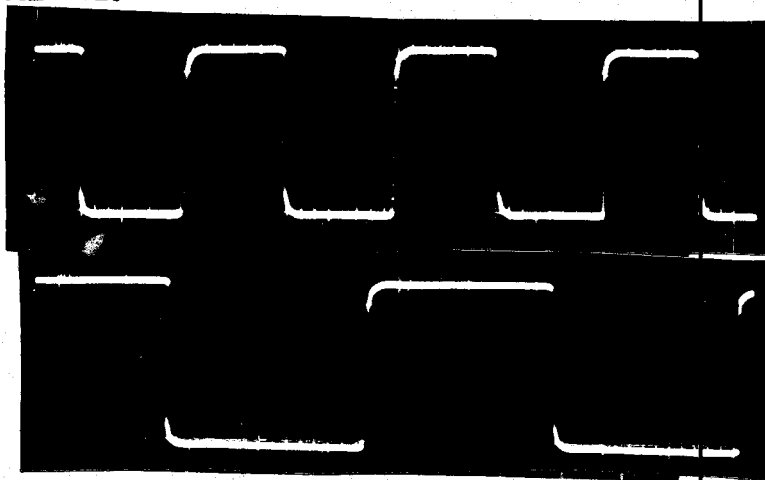
Tekanan (mmHg)	Frekuensi(Hz)	Tekanan (mmHg)	Frekuensi(Hz)
0	660790	160	496200
10	649060	170	486070
20	638740	180	476710
30	628740	190	466800
40	617490	200	457240
50	607000	210	447370
60	596510	220	437920
70	586310	230	428160
80	576140	240	418960
90	566040	250	409560
100	555810	260	400070
110	545780	270	392280
120	535730	280	380760
130	525780	290	371210
140	515820	300	361220
150	505930		

Tester: Leader LDC-823A Digital Counter

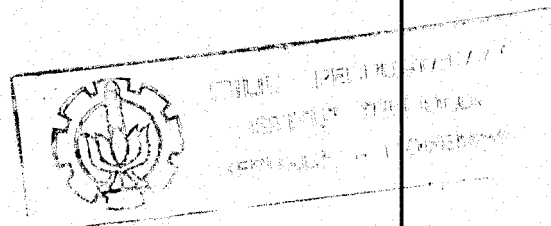


Gambar 5.1 Grafik Tekanan Terhadap Frekuensi

Bentuk sinyal output astable multivibrator ini terlihat pada foto di gambar 5.2.



Gambar 5.2 Bentuk Sinyal Output Astable Multivibrator pada Tekanan 0 mmHg (gambar atas) dan 300 mmHg (gambar bawah)



Osilloscope yang digunakan Digital Storage Oscilloscope KIKUSUI DSS5020 dengan skala time/div = 0,5 uS, dan skala volt/div = 2 V, dari ketentuan ini dapat dihitung frekuensi sinyal gambar atas, yaitu:

$$F = 1 / (5,6 \times 0,5 \text{ uS}) = 357142 \text{ Hz.}$$

Sedangkan tegangan outputnya $V = 2,5 \times 2 \text{ V} = 5 \text{ Volt.}$

Untuk sinyal pada gambar bawah:

$$F = 1 / (3,0 \times 0,5 \text{ uS}) = 666667 \text{ Hz}$$

Sedangkan tegangan outputnya $V = 2,5 \times 2 \text{ V} = 5 \text{ Volt.}$

V.2. BAGIAN TRANSDUCER BUNYI

V.2.1. Transducer Piezoelektrik

Digunakan piezoelektrik transducer dengan hasil pengukuran rata-rata, kapasitansinya 15 nF.

Amplifier yang digunakan mempunyai impedansi input 330000 Ω , sehingga frekuensi cutoff bawahnya :

$$F_{cl} = 1 / 2\pi \cdot R \cdot C$$

$$F_{cl} = 1 / 2\pi \cdot 330000 \cdot 15e^{-9}$$

$$F_{cl} = 32 \text{ Hz}$$

V.2.2. Amplifier

Digunakan Non Inverting Amplifier, dari hasil pengukuran pada level tegangan input, $V_{in} = 0,1 \text{ Volt Vpp}$, didapat: beasar tegangan output, $V_{out} = 16,5 \text{ Volt Vpp}$, sehingga penguatannya, $A = 165$

Frekuensi cutoff bawah tidak ditemukan, frekuensi cutoff atas:

$F_{ch} = 100 \text{ KHz}$, yaitu pada saat penguatan $A = 0,7 \times 165 = 115,5$

V.2.3. Filter

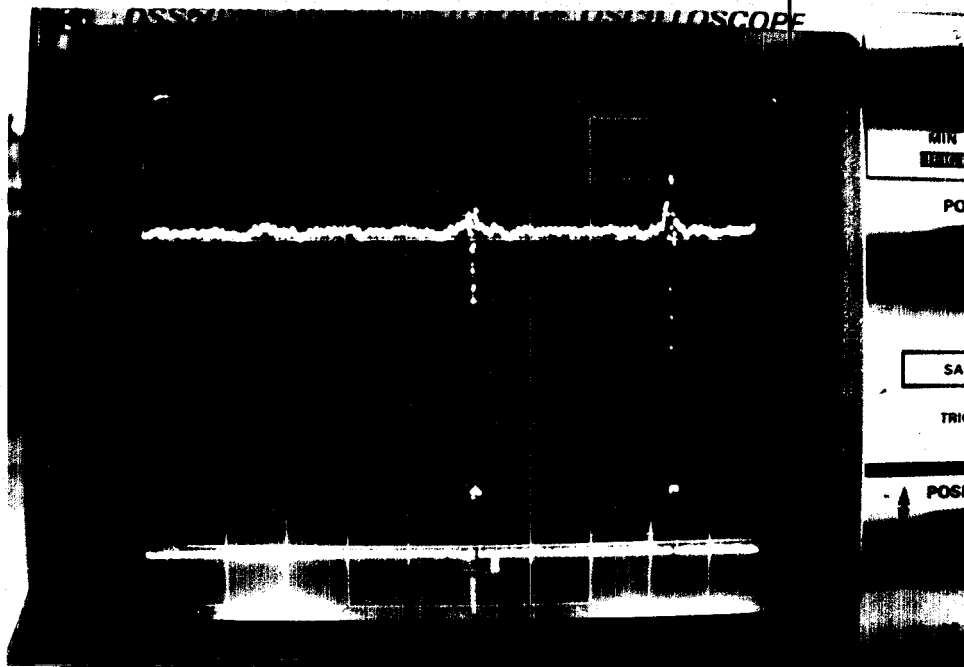
Bandpass filter yang terdiri atas lowpass disusul dengan highpass filter (masing-masing -40 dB / dec dan 40 dB / dec),

Dari hasil pengukuran:

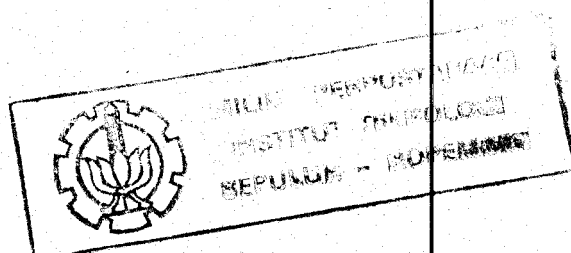
Frekuensi cutoff bawah, $F_{cl} = 22 \text{ Hz}$

Frekuensi cutoff atas, $F_{ch} = 310 \text{ Hz}$

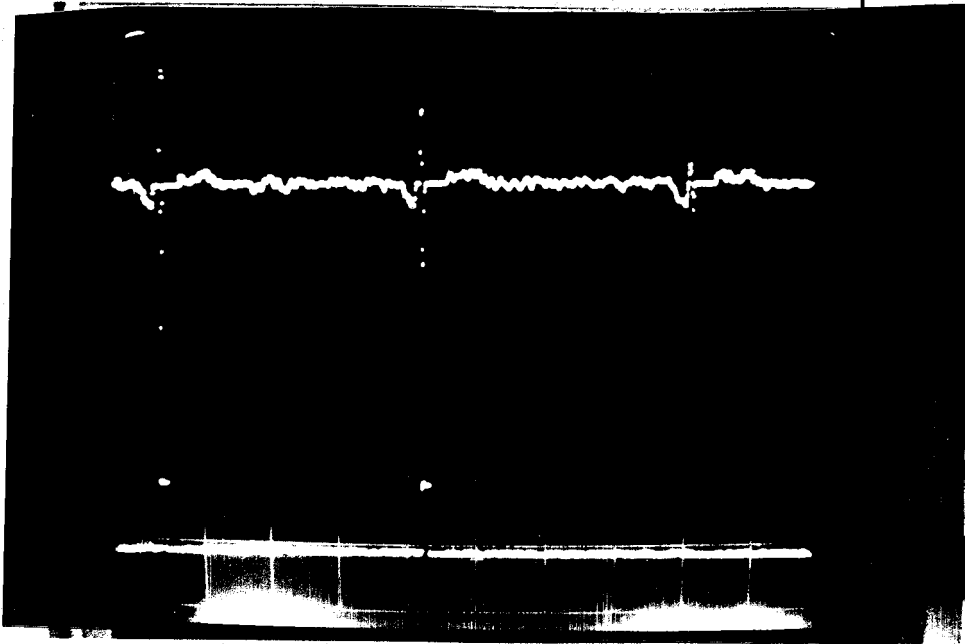
V.2.4. Sinyal Korotkoff dan Output Monostable Multivibrator



Gambar 5.3 Output Filter (atas) dan Monostable (bawah) Saat Pulsa Sistolik Terdeteksi



Output dari bagian filter dan Monostable untuk pulsa sistolik dan diastolik, diperlihatkan pada gambar 5.3 dan 5.4



Gambar 5.4 Output Filter (atas) dan Monostable (bawah) Saat Pulsa Diastolik terdeteksi

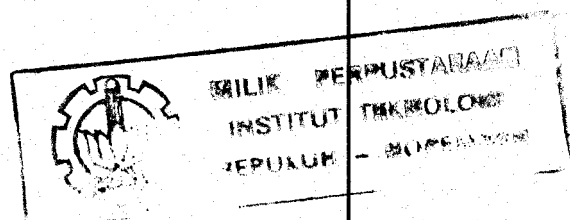
Osilloscope yang digunakan Digital Storage Oscilloscope KIKUSUI DSS5020 dengan skala time/div = 0,2 S, dan skala volt/div = 2 V (untuk output filter) dan skala volt/div = 5 V (untuk output Monostable).

Dari foto diatas dapat dihitung heartrate, hr sebagai berikut:

$$hr = 1 / (3,8 \times 0,2) \text{ Hz} = 1.316 \text{ Hz, atau } 79 \text{ beat / menit.}$$

V.3. BAGIAN PNEUMATIS

Bagian ini bertugas mengadakan tekanan ke lengan, dengan cara menaikkan tekanan cuff dan menurunkan tekanan secara bertahap. Dari



pengukuran didapat:

Waktu yang diperlukan pompa untuk memberi tekanan cuff hingga 300 mmHg

$T_{inc} = 12$ detik

Kecepatan penurunan tekanan cuff rata-rata (tekanan cuff turun karena udara keluar lewat penahan aliran udara), $V_{dec} = 4.5$ mmHg / detik

V. 4. PENGUKURAN TEKANAN

Setelah bagian transducer tekanan dihubungkan dengan sistem mikrokomputer, dan digunakan untuk mengukur tekanan maka diperoleh hasil, sebagai berikut:

Tabel 5.3. Hasil Pengukuran Tekanan

STANDARD	DATA	ERROR (%)	STANDARD	DATA	ERROR (%)
P (mmHg)	P (mmHg)		P (mmHg)	P (mmHg)	
0	1.0	-	160	161.8	1.1
10	11.0	10.0	170	171.8	1.1
20	21.2	6.0	180	182.0	1.1
30	32.4	8.0	190	191.8	0.9
40	42.0	5.0	200	202.2	1.1
50	52.2	4.4	210	211.8	0.9
60	61.8	3.0	220	222.2	1.0
70	72.4	3.4	230	231.8	0.8
80	82.0	2.5	240	242.0	0.8
90	92.0	2.2	250	252.2	0.9
100	101.8	1.8	260	263.4	1.3
110	112.0	1.8	270	272.4	0.9
120	121.6	1.3	280	282.6	0.9
130	132.2	1.7	290	291.8	0.6
140	142.0	1.4	300	299.2	0.3
150	151.8	1.2			

Tester: Nova Presameter Air Raksa

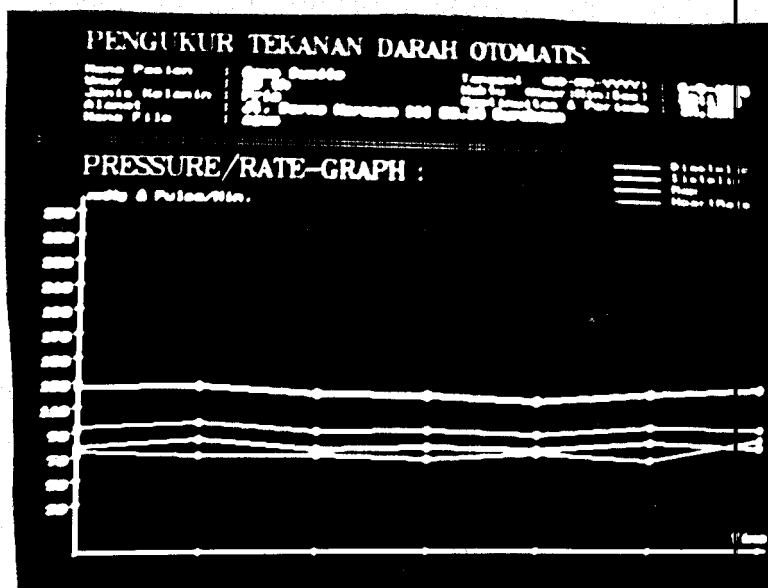
V.5. UJI COBA ALAT

Hasil uji coba alat dapat dilihat pada tabel 5.4.

Tabel 5.4. Hasil Pengukuran Tekanan Darah Seorang Sukarelawan

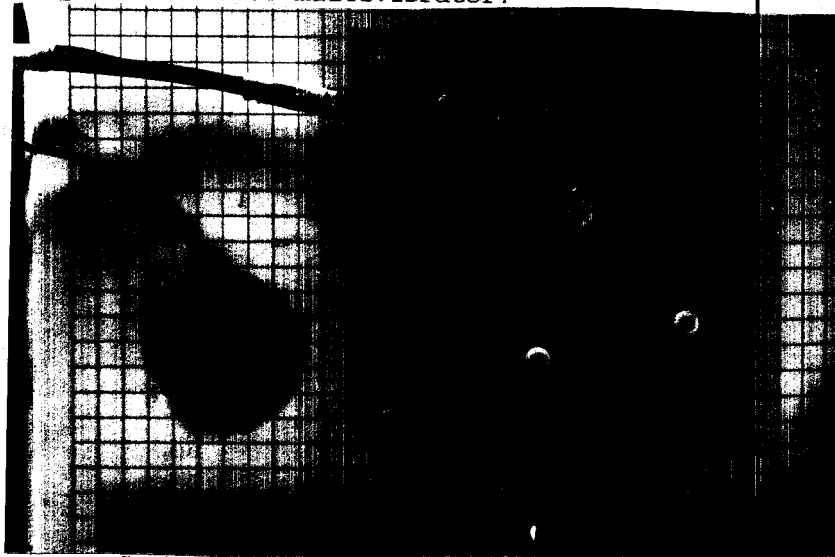
No.	Sistolik (mmHg)	Diastolik (mmHg)	M A P (mmHg)	Heart Rate (mmHg)	Lama Pengukuran (detik)
1	123	64	83	73	22.20
2	115	74	87	72	20.21
3	117	80	92	76	19.80
4	128	71	90	73	21.84
5	126	74	91	70	20.36

Uji coba alat dengan program komputer menghasilkan tampilan seperti pada gambar 5.5.

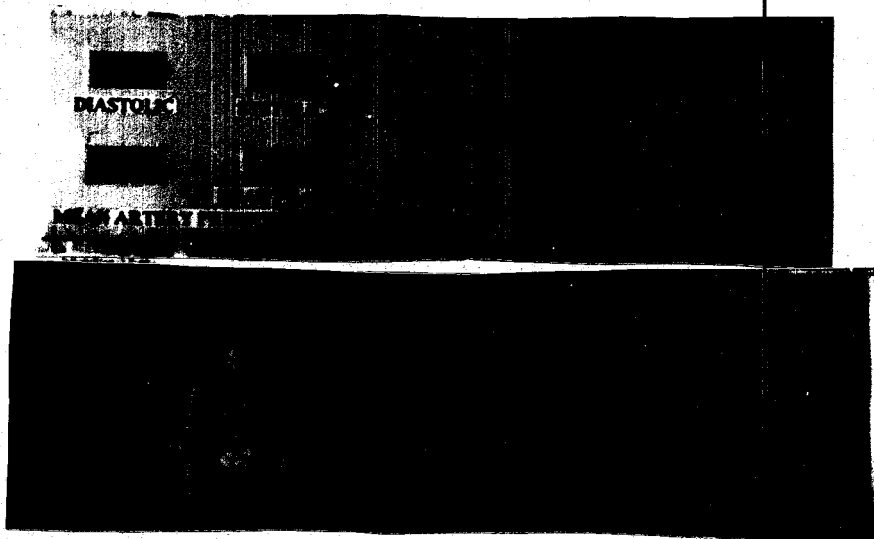


Gambar 5.5 Tampilan Grafik Hasil Pengukuran

Gambar 5.6 menunjukkan bentuk transducer yang dipakai, sebelah kiri transducer piezoelektrik dan sebelah kanan transducer tekanan beserta rangkaian astable multivibrator.

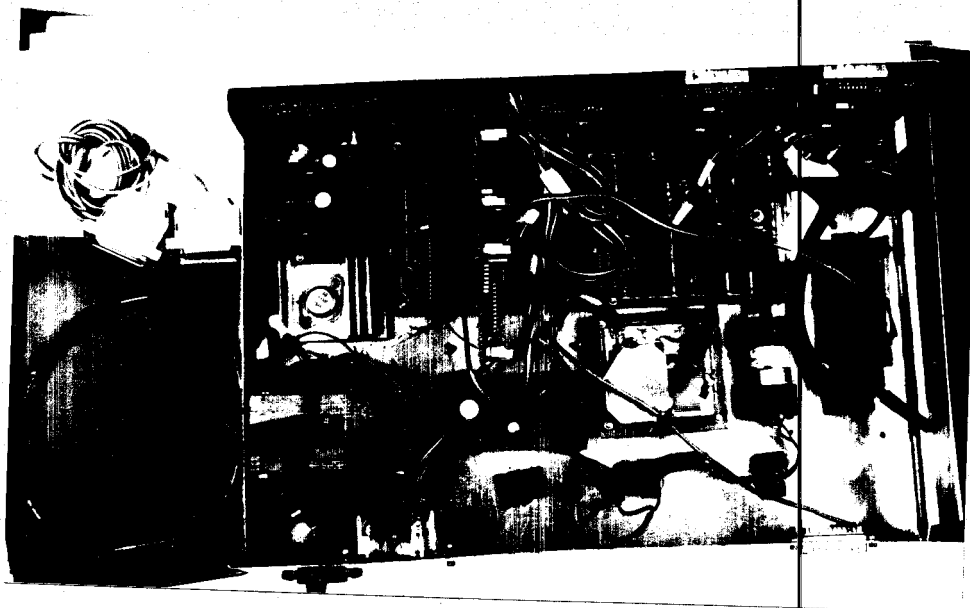


Gambar 5.6 Transducer Bunyi dan Tekanan

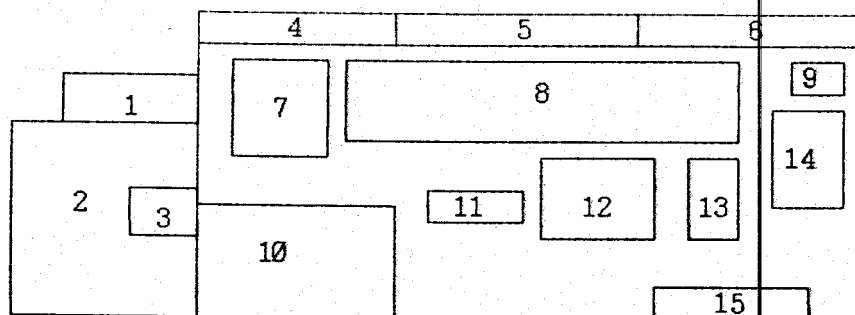


Gambar 5.7 Tampak Depan dan Belakang Alat

Gambar 5.8 menunjukkan foto alat yang dibuat beserta bagian-bagiannya



Gambar 5.8 Pengukur Tekanan Darah



Keterangan:

1. Konektor komunikasi serial RS 232
2. Cuff
3. Piezoelektrik
4. Keyboard dan speaker
5. Display alfa numerik
6. Display numerik 3 digit
7. Rangkaian catu daya
8. Rangkaian mikrokomputer
9. Katup pengeluaran dan penahan aliran udara keluar
10. Trafo daya
11. Rangkaian buffer daya untuk pompa
12. Transducer tekanan dan rangkaiannya
13. Pompa udara
14. Rangkaian amplifier, filter, komparator, dan monostable
15. Soket DB 25

BAB VI

PENUTUP

VI.1. KESIMPULAN

Dari pengamatan selama analisa untuk perencanaan, disain hardware maupun software, pembuatan, pengukuran dan uji coba, dapat diambil beberapaa kesimpulan:

1. Disain hardware dan software untuk suatu alat dilakukan dengan menganalisa kerja sistem untuk suatu maksud tertentu, didalamnya termasuk fungsi - fungsi yang harus ada, realisasi fungsi - fungsi tersebut, kombinasi fungsi secara efisien dan efektif dalam membentuk sistem, dan fleksibilitas pengembangan sistem.
2. Kombinasi kapasitansi transducer tekanan dan astable multivibrator yang menghasilkan perubahan frekuensi output terhadap input tekanan cukup baik dipakai sebagai pengukur tekanan.
3. Transducer piezoelektrik sangat baik untuk mendeteksi bunyi Korotkoff karena mempunyai respon frekuensi rendah yang baik.
4. Untuk mengatasi gangguan dalam mendeteksi bunyi Korotkoff digunakan bandpass filter, sebab bunyi Korotkoff mempunyai frekuensi sekitar 20 Hz sampai 300 Hz. Gangguan dari frekuensi rendah misalnya gerakan lengan pasien. sedangkan gangguan dari frekuensi tinggi misalnya sinyal RF (Radio Frekuensi) dari pemancar radio / TV.
5. Disain bagian hardware yang mempunyai fleksibilitas tinggi terhadap software sangat menguntungkan untuk penyempurnaan kemampuan sistem dan menyederhanakan hardware itu sendiri.
6. Peralatan dirancang dapat mengirimkan data ke komputer, sebab

pengolahan data lebih lengkap dan lanjut sangat mudah dilakukan dengan piranti tersebut.

7. Masuknya data standard, misalnya data medis sangat membantu dalam pengambilan keputusan secara cepat.
8. Secara umum pembuatan suatu peralatan dimulai dari analisa alat, disain alat, realisasi disain, dan akhirnya penyempurnaan serta pengembangannya yang mempengaruhi disain alat untuk pembuatan selanjutnya.

VI.2. SARAN

1. Untuk meningkatkan kemampuan alat dapat dilakukan dengan menyempurnakan softwarenya.
2. Adanya data medis standard memudahkan dalam pengambilan keputusan tetapi harus didukung dengan kehandalan alat dalam akusisi data.
3. Pengambilan keputusan yang bijak tidak hanya ditentukan oleh beberapa kriteria saja, makin banyak kreteria akan makin tepat keputusannya. Pada alat pengukur tekanan darah ini, keputusan hanya didasarkan pada umur dan jenis kelamin saja, untuk penyempurnaan dalam pengambilan keputusan dapat dimasukkan data lain misalnya fluktuasi tekanan terhadap waktu sehari semalam.

DAFTAR PUSTAKA

1. Coughlin, Robert F., Frederick F. Driscoll, dan Herman W. Soemitro
(penerjemah), PENGUAT OPERASIONAL DAN RANGKAIAN TERPADU
LINEAR, Erlangga, Jakarta, 1985.
2. Hall, Douglas V., MICROPROCESSOR AND INTERFACING : Programming and
Hardware, McGraw-Hill Inc, Singapore, 1986.
3. Hartono Partoharsodjo, TUNTUNAN PRAKTIS PEMROGRAMANAN BAHASA
ASSEMBLY, Pt. Elex Media Komputindo, Jakarta, 1990.
4. Tompkins, William J. and John G. Webster, DESIGN of
MICROCOMPUTER-BASED MEDICAL INSTRUMENTATION, Prentice-Hall,
Englewood cliff, New Jersey, 1981.
5. Webster, John G., (editor), MEDICAL INSTRUMENTATION : Application
and design, Houghton Mifflin Company, Boston, 1978.
6. -----, CMOS1 DIGITAL Data Dictionary and Comparison Table,
Binatronika, Jakarta, 1984.
7. -----, Data Praktis Elektronika, PT. Elex Media Komputindo,
Jakarta, 1987.
8. -----, IC DIGITAL TTL Data and Comparison Table, Binatronika,
Jakarta, 1984.
9. -----, MICROPROCESSOR AND PERIPHERAL HANDBOOK I, Intel Corp.
Literature Distribution, Santa Clara, 1988.
10. -----, MICROPROCESSOR AND PERIPHERAL HANDBOOK II, Intel Corp.
Literature Distribution, Santa Clara, 1988.

DATA PENGUKURAN FREKUENSI BAGIAN TRANSDUCER TEKANA

TEKANAN P (mmHg)	FREKUENSI TERUKUR					
	F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)	F5 (Hz)	F (Hz)
0	661750	660780	660800	660810	660810	660790
10	649070	660480	645550	651040	649150	649060
20	638030	638970	638030	638990	639660	638740
30	628870	628720	628710	628780	628640	628740
40	617050	617480	617370	617980	617580	617490
50	606850	606850	607050	606950	607300	607000
60	596350	596750	596810	596050	596580	596510
70	586020	585930	585740	586970	586900	586310
80	576030	575970	576160	576630	575900	576140
90	565650	565460	565800	566500	566810	566040
100	555890	555690	555750	555630	556070	555810
110	546060	545710	545980	545650	545490	545780
120	536080	535750	535600	535590	535650	535730
130	525780	525660	525870	525900	525700	525780
140	515740	515690	515970	515980	515700	515820
150	505980	505870	505980	505830	505980	505930
160	495680	495980	496370	496770	496180	496200
170	485790	486280	485920	486590	485780	486070
180	476350	476830	476530	477030	476800	476710
190	466530	466980	466850	466780	466840	466800
200	456730	456890	456930	457680	457790	457240
210	446540	447580	447500	447280	447970	447370
220	436960	437520	438370	438510	438250	437920
230	426770	428650	428370	428470	428530	428180
240	417850	419030	419850	419130	418950	418960
250	408960	409650	409420	409840	409950	409560
260	399540	399980	400570	400380	399860	400070
270	398890	390800	390150	390570	390980	392280
280	380830	380840	380590	380970	380580	380760
290	370780	371090	371680	371530	370970	371210
300	360650	360590	361970	361030	361850	361220

TESTER: Leader LDC 823A Digital Counter

PENGUKURAN TEKANAN

STANDARD	HASIL PENGUKURAN OLEH ALAT					
P (mmHg)	P1 (mmHg)	P2 (mmHg)	P3 (mmHg)	P4 (mmHg)	P5 (mmHg)	\bar{P}
0	1	1	1	1	1	1.0
10	11	10	11	11	12	11.0
20	21	21	22	21	21	21.2
30	32	33	33	32	32	32.4
40	42	42	42	42	42	42.0
50	52	52	53	52	52	52.2
60	62	62	62	62	61	61.8
70	73	72	72	72	73	72.4
80	82	82	82	82	82	82.0
90	92	92	93	91	92	92.0
100	102	101	102	102	102	101.8
110	112	112	112	112	112	112.0
120	122	121	122	121	122	121.6
130	133	132	132	132	132	132.2
140	142	142	142	142	142	142.0
150	152	152	152	151	152	151.8
160	162	162	162	162	161	161.8
170	172	172	172	172	171	171.8
180	182	182	182	182	182	182.0
190	192	192	191	192	192	191.8
200	202	202	203	202	202	202.2
210	212	212	212	212	211	211.8
220	223	222	222	222	222	222.2
230	232	231	232	232	232	231.8
240	243	241	243	242	241	242.0
250	252	252	253	252	252	252.2
260	264	263	263	263	264	263.4
270	273	271	273	273	272	272.4
280	282	282	283	283	283	282.6
290	291	291	293	292	292	291.8
300	299	299	300	299	299	299.2

Instrumen standar: Nova Presameter Air raksa

```

uses graph,dos,crt;

procedure RSWrite(data:byte);
var regs:registers;
begin
    regs.ah:=$1;
    regs.dx:=$01;
    regs.al:=data;
    intr($14,regs);
end;

label skip,loadit,makeit,dograph,beginagain,askagain,enditall,
sinkron1,sinkron2,newprocess,skiptoload1,skiptoload2,
skiptoload3,skiptoload4,skiptime,makeit1,gosync;
testcomlink1,testcomlink2,comlinkfail,comlinkready;

var
    data:byte;
    cdata:char;
    regs:registers;
    nama,umur,jkel,addr:string;
    nfil:string;
    filex:text;
    gdriver,gmode,loadval:integer;
    pressx1,pressxlalu1,pressx2,pressxlalu2,pressx3,
    pressxlalu3:integer;
    press1,press2,press3,heart:integer;
    hratex,hratexlalu:integer;
    konth,kont,periode:longint;
    skont,speriode:string;
    xmax,a,b,blalu:longint;
    c:longint;
    cstr:string[3];
    pressure1:array[0..300] of integer;
    pressure2:array[0..300] of integer;
    pressure3:array[0..300] of integer;
    heartrate:array[0..300] of integer;
    year,month,day,dayofweek,hour,minute,second,secl00:word;
    smonth,sday,sdayofweek,shour,sminute,ssecond,
    ssecl00:string[2];
    syear:string[4];
    date,time:string;
    inkey:char;

begin
beginagain:
clrscr;
gdriver:=detect;
initgraph(gdriver,gmode,'');
settextstyle(1,0,4);
outtextxy(27,50,'AUTOMATIC BLOOD PRESSURE MONITOR.');
```

```

line(27,90,625,90);
{Inisialisasi RS}
settextstyle(1,0,2);
line(40,200,40,300);
line(37,200,37,300);

```

```

loadval:=0;
outtextxy(50,150,'(L)oad, (N)ew or (Q)uit . . .');
askagain:
inkey:=readkey;
if inkey='Q' then goto EnditAll;
if inkey='q' then goto EnditAll;
if inkey='L' then goto Loadit;
if inkey='l' then goto Loadit;
if inkey='N' then goto Makeit1;
if inkey='n' then goto Makeit1 else goto askagain;

makeit1:
outtextxy(50,200,'Initializing RS-Driver.');
```

regs.al:=\$23;	{:00100011 Baud150, Noparity, StopBit1,
	8bitData}

```

    regs.ah:=$0;
    regs.dx:=$01;
    intr($14,regs);
outtextxy(50,220,'Ok.');
```

{Check Comline}

```

outtextxy(50,240,'Testing Transmission Integrity.');
```

testcomlink1:

```

    regs.ah:=$2;
    regs.dx:=$01;
    intr($14,regs);
    if regs.al<>36 then goto testcomlink1;
    cdata:='#';
    data:=ord(cdata);
    rswrite(data);
    regs.al:=0;
    a:=0;
```

testcomlink2:

```

    inc(a);
    if a>=11 then goto comlinkfail;
    regs.ah:=$2;
    regs.dx:=$01;
    intr($14,regs);
    if regs.al<>36 then goto testcomlink2;
    goto comlinkready;
```

comlinkfail:

```

    outtextxy(50,260,'ComLink Fail at Test2, Please check circuit
and wiring.');
```

goto enditall;

comlinkready:

```

    outtextxy(50,260,'Comlink Test done, Transmission line is
good.');
```

outtextxy(50,280,'<ENTER> when ready.');

```

    inkey:=readkey;
    if inkey=' ' then goto comlinkready;
```

closegraph;

goto makeit;

```

Loadit:
LoadVal:=1;
closegraph;
writeln('Enter Filename to be loaded: . . .');
readln(nfil);
assign(filex,'C:\'+nfil);
reset(filex);
readln(filex,nama);
readln(filex,umur);
readln(filex,jkel);
readln(filex,addr);
readln(filex,kont);
readln(filex,periode);
konth:=kont*60;
for a:=0 to (konth div periode) do
begin
readln(filex,pressure1[a]);
readln(filex,pressure2[a]);
readln(filex,pressure3[a]);
readln(filex,heartrate[a]);
end;
goto dograph;

```

```

Makeit:
write('Nama Pasien   =');
readln(nama);
write('Umur          =');
readln(umur);
write('Jenis Kelamin =');
readln(jkel);
write('Alamat        =');
readln(addr);
write('Nama file     =');
readln(nfil);

```

```

DoGraph:
gdriver:=detect;
initgraph(gdriver,gmode,'');
setcolor(9);
rectangle(0,0,639,100);
rectangle(0,104,639,479);
setfillstyle(9,1);
bar(40,160,635,400);
setfillstyle(0,0);
setcolor(12);
outtextxy(550,110,'Diastolic');
outtextxy(550,120,'Sistolic');
outtextxy(550,130,'Map');
outtextxy(550,140,'HeartRate');
setcolor(13);
line(500,115,540,115);
setcolor(14);
line(500,125,540,125);
setcolor(11);

```

```

line(500,135,540,135);
setcolor(7);
line(500,145,540,145);
settextstyle(1,0,3);
setcolor(12);
outtextxy(40,5,'PENGUKUR TEKANAN DARAH OTOMATIS.');
```

PRESSURE/RATE-GRAPH :

```

settextstyle(0,0,1);
setcolor(10);
outtextxy(40,40,'Nama Pasien  :');
outtextxy(40,50,'Umur        :');
outtextxy(40,60,'Jenis Kelamin :');
outtextxy(40,70,'Alamat       :');
outtextxy(40,80,'Nama File    :');
outtextxy(365,40,'Tanggal   (DD-MM-YYYY) :');
outtextxy(365,50,'Waktu     (Hour:Min:Sec) :');
outtextxy(365,60,'Kontinuitas & Periode :');
setcolor(15);
outtextxy(170,40,nama);
outtextxy(170,50,umur);
outtextxy(170,60,jkel);
outtextxy(170,70,addr);
outtextxy(170,80,nfil);
setcolor(14);
outtextxy(45,145,'mm');
setcolor(13);
outtextxy(61,145,'H');
setcolor(11);
outtextxy(69,145,'g');
setcolor(7);
outtextxy(100,145,'Pulse/Min.');
```

&

```

setcolor(15);
outtextxy(85,145,'&');
outtextxy(605,425,'Time');
```

```

for a:=1 to 13 do
begin
  b:=a*20;
  setcolor(15);
  line(37,430-10-b,42,430-10-b);
  setcolor(15);
  c:=30+(a-1)*20;
  str(c,cstr);
  if length(cstr)=2 then cstr:=' '+cstr;
  outtextxy(10,430-10-b,cstr);
end;
line(40,440,635,440);
line(40,150,40,440);

if loadval=1 then goto skiptoload1;

regs.al:=0;
(Data serial untuk kontinuitas dan periode)
sinkron1:
  regs.ah:=$2;
```



```

        regs.dx:=$01;
        intr($14,regs);
        if regs.al<>36 then goto sinkron1;
cdata:='#';
data:=ord(cdata);
rswrite(data);
        regs.ah:=$2;
        regs.dx:=$01;
        intr($14,regs);
kont:=regs.al;
konth:=kont*60;
        regs.ah:=$2;
        regs.dx:=$01;
        intr($14,regs);
periode:=regs.al;

skiptoload1:
str(kont,skont);
str(periode,speriode);
outtextxy(560,60,skont+'H,'+speriode+'M');
xmax:=(590 * periode) div konth;

if loadval=1 then goto skiptoload2;
assign(filex,'C:\'+nfil);
rewrite(filex);
writeln(filex,nama);
writeln(filex,umur);
writeln(filex,jkel);
writeln(filex,addr);
writeln(filex,kont);
writeln(filex,periode);
close(filex);
skiptoload2:

b:=0;
pressx1:=0;
pressx2:=0;
pressx3:=0;
date:='';
time:='';

{ M A I N   L O O P   !!!!!!!!!!!!!!!}
for a:=0 to (konth div periode) do
begin
press1:=0;
press2:=0;
press3:=0;
heart:=0;
if loadval=1 then goto skiptoload3;

(Masukkan data serial untuk Diastolic, Sistolic, Map, Heartrate)
sinkron2:
        if keypressed=false then goto gosync;
        inkey:=readkey;
        if inkey='Q' then

```

```

begin
  setcolor(15);
  outtextxy(365,80,'Are you Sure <Y/N> ??');
  inkey:=readkey;
  if inkey='Y' then goto newprocess;
  if inkey='y' then goto newprocess;
  setcolor(0);
  outtextxy(365,80,'Are you Sure <Y/N> ??');
  goto gosync;
end;
if inkey='q' then
begin
  setcolor(15);
  outtextxy(365,80,'Are you Sure <Y/N> ??');
  inkey:=readkey;
  if inkey='Y' then goto newprocess;
  if inkey='y' then goto newprocess;
  setcolor(0);
  outtextxy(365,80,'Are you Sure <Y/N> ??');
  goto gosync;
end;
gosync:
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  if regs.al=0 then goto sinkron2;
  if regs.al<>36 then goto sinkron2;
  cdata:='#';
  data:=ord(cdata);
  rswrite(data);
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  press1:=regs.al;
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  pressure1[a]:=regs.al;
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  press2:=regs.al;
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  pressure2[a]:=regs.al;
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  press3:=regs.al;
  regs.ah:=$2;
  regs.dx:=$01;
  intr($14,regs);
  pressure3[a]:=regs.al;
  regs.ah:=$2;

```

```

        regs.dx:=$01;
        intr($14,regs);
        heart:=regs.al;
        regs.ah:=$2;
        regs.dx:=$01;
        intr($14,regs);
        heartrate[a]:=regs.al;

skiptoload3:
        blalu:=b;
        b:=xmax*a;
        getdate(year,month,day,dayofweek);
        gettime(hour,minute,second,secl00);
        str(year,syear);
        str(month,smonth);
        str(day,sday);
        str(hour,shour);
        str(minute,sminute);
        str(second,ssecond);
        setcolor(0);
        outtextxy(560,40,date);
        outtextxy(560,50,time);
        setcolor(15);
        line(40+b,438,40+b,442);
        date:=sday+'-'+smonth+'-'+syear;
        time:=shour+':'+sminute+':'+ssecond;
        outtextxy(560,40,date);
        outtextxy(560,50,time);
        if loadval=1 then goto skiptime;
        if a=0 then
            begin
                outtextxy(40+b,450,time);
                line(40+b,442,40+b,448);
            end;
        if a=((konth div periode) div 2) then
            begin
                outtextxy(40+b,450,time);
                line(40+b,442,40+b,448);
            end;
        if a=(konth div periode) then
            begin
                outtextxy(40+b-50,450,time);
                line(40+b,442,40+b,448);
            end;
skiptime:
        pressure1[a]:=(press1*200)+pressure1[a];
        pressure2[a]:=(press2*200)+pressure2[a];
        pressure3[a]:=(press3*200)+pressure3[a];
        heartrate[a]:=(heart*200)+heartrate[a];
        pressxlalul:=pressx1;
        pressx1:=400-(pressure1[a]-30);
        pressxlalu2:=pressx2;
        pressx2:=400-(pressure2[a]-30);
        pressxlalu3:=pressx3;
        pressx3:=400-(pressure3[a]-30);

```

```

hratexlalu:=hratex;
hratex:=390-(heartrate[a]-40);
setcolor(13);
circle(40+b,pressx1,2);
setcolor(14);
circle(40+b,pressx2,2);
setcolor(11);
circle(40+b,pressx3,2);
setcolor(7);
circle(40+b,hratex,2);
if a=0 then goto skip;
setcolor(13);
line(40+b,pressx1,40+blalu,pressxlalu1);
setcolor(14);
line(40+b,pressx2,40+blalu,pressxlalu2);
setcolor(11);
line(40+b,pressx3,40+blalu,pressxlalu3);
setcolor(7);
line(40+b,hratex,40+blalu,hratexlalu);
setcolor(15);
skip:
  if loadval=1 then goto skiptoload4;

  assign(filex,'C:\'+nfil);
  reset(filex);
  writeln(filex,pressure1[a]);
  writeln(filex,pressure2[a]);
  writeln(filex,pressure3[a]);
  writeln(filex,heartrate[a]);
  close(filex);

skiptoload4:
  end;
readln;
newprocess:
closegraph;
goto beginagain;
enditall:
end.

```

```

;=====
;1. Equalisation
;=====
port1_a      equ 300h
port1_b      equ 301h
port1_c      equ 302h
cword1       equ 303h
port2_a      equ 304h
port2_b      equ 305h
port2_c      equ 306h
cword2       equ 307h
counter0     equ 308h
counter1     equ 309h
counter2     equ 30ah
cwordc       equ 30bh
silent       equ 0b0h           ;mode0
U_control_add equ 30dh
* U_data_add  equ 30ch

;=====
;2. Beginning of Eprom
;=====
org 0

;=====
;3. Space for vector interrupt
;=====
start      db 100h dup(0ffh)
org 100h

;=====
;4. Memory booking
;=====
pra        db 0
transmited db ?
received   db ?
t_low      dw 0
t_high     dw 3
f          db 0      ;after interupt/nmi f = 1
freq       dw 0      ;pulsa/10ms
pressure   dw 0      ;pressure value ( x mmHg )
f1         dw 0
f2         dw 0
p1         dw 0
p2         dw 0
key_data   db 'C840D951EA62FB73'
geser      db 11111110b
pb         db ?
pc         db ?
locus      dw ?
reference   dw ?
expression dw ?
frametime  dw ?
oldtime    dw ?
newtime    dw ?

```

pduration	dw ?
tpduration	dw ?
totalpulse	db ?
first	db ?
t	dw ?
fsys	dw ?
fdias	dw ?
sysp	dw 1
diasp	dw 1
map	dw 1
hr	dw 1
syspt1	db ?
syspt2	db ?
diaspt1	db ?
diaspt2	db ?
mapt1	db ?
mapt2	db ?
hrt1	db ?
hrt2	db ?
p150	dw 5059
p160	dw 4962
p170	dw 4861
p180	dw 4767
p190	dw 4668
p200	dw 4572
p210	dw 4473
p220	dw 4379
p230	dw 4282
p240	dw 4190
p250	dw 4096
p260	dw 4001
p270	dw 3923
p280	dw 3808
p290	dw 3712
p300	dw 3612
monfail	db ?
aircharge	dw ?
key	db ?
keypressed	db 0
dmode	db ?
dpressure	dw ?
dage	db ?
dsex	db ?
dperiode	dw ?
dcontinuity	dw ?
dalarm	db ?
ddiagnose	db ?
dcomlink	db ?
try	db 3
dt_low	dw ?
dt_high	dw ?
sys_low	dw ?
sys_high	dw ?
dias_low	dw ?
dias_high	dw ?

```

decflag      db 00000000b ;according to set diaslow,diashigh,
               ;syslow,syshigh,4 last bit not used
failflag     db 0          ;failcom caused flag set

```

```

;[Diagnose / Word / Character] decoder

```

```

com_test db 0d8h,0fh,0c0h,0fh,24h,0fh,0ffh,0fh,0fbh,07h,0d8h,0bh
          db 0c9h,0bh,0fbh,07h
fail      db 0f8h,0bh,0e0h,0bh,0dbb,07h,0dch,0fh,8 dup (0ffh)
pass      db 0f0h,0bh,0e0h,0bh,0c9h,0bh,0c9h,0bh,8 dup (0ffh)
pset      db 0f0h,0bh,0f0h,0ah,0d8h,0bh,0c9h,0bh,0c9h,0bh,0c4h,0fh
          db 0f0h,0ah,0d8h,0bh
mode      db 24h,0fh,0c0h,0fh,0c3h,07h,0d8h,0bh,0ffh,0fh,0ffh,0fh
          db 0ffh,0fh,0ffh,0fh
age       db 0e0h,0bh,0c8h,0bh,0d8h,0bh,0ffh,0fh,0ffh,0fh,0ffh,0fh
          db 0ffh,0fh,0ffh,0fh
sex       db 0c9h,0bh,0d8h,0bh,3fh,0ch,0ffh,0fh,0ffh,0fh,0ffh,0fh
          db 0ffh,0fh,0ffh,0fh
periode   db 0f0h,0bh,0d8h,0bh,0f0h,0ah,0dbb,07h,0c0h,0fh,0c3h,07h
          db 0d8h,0bh,0ffh,0fh
continuity db 0c3h,07h,0c4h,0fh,0f0h,0ah,0e0h,0bh,0fbh,07h,0dbb
          db 07h,0c0h,0fh,0a4h,0eh
alarm     db 0e0h,0bh,0dch,0fh,0e0h,0bh,0f0h,0ah,24h,0fh,0ffh,0fh
          db 0ffh,0fh,0ffh,0fh
diagns    db 0c3h,07h,0dbb,07h,0e0h,0bh,0c8h,0bh,0a4h,0eh,0c9h,0bh
          db 0ffh,0fh,0ffh,0fh
ok        db 0c0h,0fh,7ch,0ch,0ffh,0fh,0ffh,0fh,0ffh,0fh,0ffh,0fh
          db 0ffh,0fh,0ffh,0fh
endx      db 0d8h,0bh,0a4h,0eh,0c3h,07h
          db 10 dup (0ffh)
comlink   db 0d8h,0fh,0c0h,0fh,24h,0fh,0dch,0fh,0dbb,07h,0a4h,0eh
          db 7ch,0ch,0ffh,0fh
getdata   db 0c8h,0bh,0d8h,0bh,0fbh,07h,0ffh,0fh,0c3h,07h,0e0h,0bh
          db 0fbh,07h,0e0h,0bh
monitorf  db 24h,0fh,0c0h,0fh,0a4h,0eh,0ffh,0fh,0f8h,0bh,0e0h,0bh
          db 0dbb,07h,0dch,0fh
          db 15 dup (0ffh),00h
normalbp  db 0a4h,0eh,0c0h,0fh,0f0h,0ah,24h,0fh,0e0h,0bh,0dch,0fh
          db 0ffh,0fh,0c3h,03h,0dch,0fh,0c0h,0fh,0c0h,0fh,0c3h,07h
          db 0ffh,0fh,0f0h,0bh,0f0h,0ah,0d8h,0bh,0c9h,0bh,0c9h,0bh
          db 0c4h,0fh,0f0h,0ah,0d8h,0bh
          db 15 dup (0ffh),00h
abnorbp   db 0e0h,0bh,0c3h,03h,0a4h,0eh,0c0h,0fh,0f0h,0ah,24h,0fh
          db 0e0h,0bh,0dch,0fh,0ffh,0fh,0c3h,03h,0dch,0fh,0c0h,0fh
          db 0c0h,0fh,0c3h,07h,0ffh,0fh,0f0h,0bh,0f0h,0ah,0d8h,0bh
          db 0c9h,0bh,0c9h,0bh,0c4h,0fh,0f0h,0ah,0d8h,0bh
          db 15 dup (0ffh),00h
syshigh   db 0c9h,0bh,0c5h,0bh,0c9h,0bh,0fbh,07h,0c0h,0fh,0dch,0fh
          db 0dbb,07h,0d8h,0fh,0ffh,0fh,0fbh,07h,0c0h,0fh,0c0h,0fh
          db 0ffh,0fh,0e4h,0bh,0dbb,07h,0c8h,0bh,0e4h,0bh
          db 15 dup (0ffh),00h
syslow    db 0c9h,0bh,0c5h,0bh,0c9h,0bh,0fbh,07h,0c0h,0fh,0dch,0fh
          db 0dbb,07h,0d8h,0fh,0ffh,0fh,0fbh,07h,0c0h,0fh,0c0h,0fh
          db 0ffh,0fh,0dch,0fh,0c0h,0fh,0e4h,0ch
          db 15 dup (0ffh),00h
diashigh  db 0c3h,07h,0dbb,07h,0e0h,0bh,0c9h,0bh,0fbh,07h,0c0h,0fh

```

```

        db 0dch,0fh,0dbb,07h,0d8h,0fh,0ffh,0fh,0fbh,07h,0c0h,0fh
        db 0c0h,0fh,0ffh,0fh,0e4h,0bh,0dbb,07h,0c8h,0bh,0e4h,0bh
        db 15 dup (0ffh),00h
diaslow db 0c3h,07h,0dbb,07h,0e0h,0bh,0c9h,0bh,0fbh,07h,0c0h,0fh
        db 0dch,0fh,0dbb,07h,0d8h,0fh,0ffh,0fh,0fbh,07h,0c0h,0fh
        db 0c0h,0fh,0ffh,0fh,0dch,0fh,0c0h,0fh,0e4h,0ch
        db 15 dup (0ffh),00h
patent  db 0e0h,0bh,0c4h,0fh,0fbh,07h,0c0h,0fh,24h,0fh,0e0h,0bh
        db 0fbh,07h,0dbb,07h,0d8h,0fh,0ffh,0fh,0c3h,0bh,0dch,0fh
        db 0c0h,0fh,0c0h,0fh,0c3h,07h,0ffh,0fh,0f0h,0bh,0f0h,0ah
        db 0d8h,0bh,0c9h,0bh,0c9h,0bh,0c4h,0fh,0f0h,0ah,0d8h,0bh
        db 0ffh,0fh,24h,0fh,0c0h,0fh,0a4h,0eh,0dbb,07h,0fbh,07h
        db 0c0h,0fh,0f0h,0ah,0ffh,0fh,7fh,0eh,0d8h,0fh,0fbh,0dh
        db 0ffh,0fh,0e0h,0bh,0c8h,0bh,0c4h,0fh,0c9h,0bh,0ffh,0fh
        db 0c9h,0bh,0c4h,0fh,0c9h,0bh,0dbb,07h,0dch,0fh,0c0h,0fh
        db 0ffh,0fh,0ffh,07h,0clh,0bh,0clh,0bh,0c3h,0bh
        db 15 dup (0ffh),00h
dat_end db 0

```

```

;=====
;5. Programme entry point
;=====

```

```

begin:

```

```

;*****

```

```

;Segment setting

```

```

;*****

```

```

cli

```

```

mov ax,0

```

```

mov ss,ax ; setting stack segment

```

```

mov bp,ax ; setting base pointer

```

```

mov ax,07f0h

```

```

mov sp,ax ; setting stack pointer

```

```

cld

```

```

mov ax,0

```

```

mov es,ax ; setting extra segment

```

```

mov ax,cs

```

```

mov ds,ax ; setting data segment

```

```

lea si,pra

```

```

mov di,si

```

```

mov cx,(dat_end-pra)

```

```

rep movsb ; copy data dari eprom ke ram

```

```

mov ax,0

```

```

mov ds,ax ; setting data segment

```

```

;*****

```

```

;Create interrupt vector table

```

```

;*****

```

```

mov ax,0

```

```

mov es,ax

```

```

mov word ptr[es:000ah],0fe00h ;segment interrupt
                                service routine

```

```

mov word ptr[es:0008h],offset nmi

```

```

;*****

```

```

;Inilitalize part

```

```

;*****

```

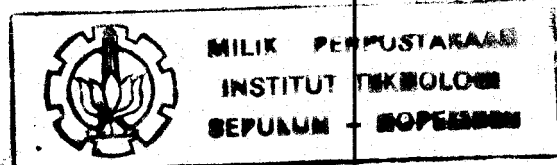


```

;8255-1
    mov     al,10001000b    ;port_c lower=output, port_b=output
    mov     dx,cword1      ;port_c upper = input
    out     dx,al          ;port_a = output, mode 0
;8255-2
    mov     al,10001000b    ;port_c lower=output, port_b=output
    mov     dx,cword2      ;port_c upper = input
    out     dx,al          ;port_a = output, mode 0
;8253-counter0
    mov     al,00110000b    ;count0, word, mode 0, binary counter
    mov     dx,cwordc
    out     dx,al
;8253-counter1
    mov     al,01110101b    ;counter1, word, mode 2, bcd counter
    mov     dx,cwordc
    out     dx,al
;8253-counter2
    mov     al,10110111b    ;counter2, word, mode 3, bcd counter
    mov     dx,cwordc
    out     dx,al
;8251-usart
    mov     dx,U_control_add ;control reg address
    mov     al,0            ;send 0's to guaranteed device
    out     dx,al          ;is in the command instruction
                                format
    mov     cx,200          ;before the reset command is
                                issued
d0:    loop    d0            ;and delay after sending each
                                command
    out     dx,al          ;instruction
    mov     cx,200
d1:    loop    d1
    out     dx,al
    mov     cx,200
d2:    loop    d2
    mov     al,01000000b    ;send internal reset command
    out     dx,al
    mov     cx,200
d3:    loop    d3
    mov     al,01001111b    ;load mode word stop bit 1, no
                                parity
    out     dx,al          ;8 bit data, baudrate factor 64x
    mov     cx,200
d4:    loop    d4
    mov     al,00110111b    ;load command word like 'douglas'
    out     dx,al          ;page 447

;*****
;NMI disable
;*****
;disable signal interrupt generator 8253
    mov     al,00000000b
    mov     dx,port1_b
    out     dx,al

```



```

;*****
;Prepare to start NMI timed interrupt
;*****
;counter 1, mode 2, generator f=500Hz, T=2ms
    mov     dx,counter1
    mov     al,09h
    out     dx,al
    mov     al,19h
    out     dx,al
;enable generator sinyal interrupt 8253
    mov     al,00001000b
    mov     dx,port1_b
    out     dx,al

;*****
;Reset numerik display
;*****
    call    dreset
;*****
;*****
;MAIN PROGRAMME
;*****
;header & copywrite expose
    lea     bx,patent
    mov     expression,bx
    call    diagnose
    call    start_sound
    lea     bx,getdata
    mov     expression,bx
    mov     keypressed,0
    call    word_expose
;start to input data
programme:
    call    data_input
    cmp     dmode,0
    je      manualmode
    jmp     automaticmode

;*****
manualmode:
    call    manualexpose
;initial pressure charged to cuff
    sub     dpressure,30
    mov     ax,dpressure
    mov     sysp,ax
    call    monitor
    call    end_sound
    lea     bx,endx
    mov     expression,bx
    mov     keypressed,0
    call    word_expose
    jmp     programme

;*****

```

```

automaticmode:
    call    autoexpose
    cmp     dcomlink,1
    jne     not_com1
    call    synchron
    mov     ax,dcontinuity
    mov     transmitted,al
    call    usart_out
    mov     ax,dperiode
    mov     transmitted,al
    call    usart_out
not_com1:
;change continuity from hour to minutes
    mov     ax,60
    mul     dcontinuity
    mov     dcontinuity,ax
;initial pressure charged to cuff
    sub     dpressure,30
    mov     ax,dpressure
    mov     sysp,ax
;initialize time var
    and     t_low,0
    and     t_high,0
continued:
;expose a mode
    call    autoexpose
;get time for determinate periode
    mov     ax,t_low
    mov     dt_low,ax
    mov     ax,t_high
    mov     dt_high,ax
;system on progress
;determinate and expose result
    call    monitor
;check com facility
    cmp     dcomlink,1
    jne     not_com2
;convert data word to 2 sending byte
    call    separate
;sending data through com
    call    synchron
    mov     al,diaspt1
    mov     transmitted,al
    call    usart_out
    mov     al,diaspt2
    mov     transmitted,al
    call    usart_out
    mov     al,syspt1
    mov     transmitted,al
    call    usart_out
    mov     al,syspt2
    mov     transmitted,al
    call    usart_out
    mov     al,mapt1
    mov     transmitted,al

```

```

        call    usart_out
        mov     al,mapt2
        mov     transmited,al
        call    usart_out
        mov     al,hrt1
        mov     transmited,al
        call    usart_out
        mov     al,hrt2
        mov     transmited,al
        call    usart_out
not_com2:
        cmp     ddiagnose,0
        je      continued1
        cmp     monfail,1
        je      continued1
        call    decision
continued1:
;continuity pass
        cmp     dcontinuity,1440
        je      fulltime
        mov     ax,dcontinuity
        cmp     t_high,ax
        jb      continued2
        jmp     complete
fulltime:
        mov     ax,t_high
        add     ax,1
        cmp     ax,dcontinuity
        jb      continued2
        jmp     complete
continued2:
        call    waitexpose
        mov     ax,dt_high
        add     ax,dperiode
continued3:
        cmp     ax,t_high
        ja      continued3
        mov     ax,dt_low
continued4:
        cmp     ax,t_low
        ja      continued4
        jmp     continued
complete:
        call    end_sound
        lea     bx,endx
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        jmp     programme
;*****
;*****

;*****
;NMI Routine
;*****

```

```

nmi    proc far
        cli
        inc     t_low
        cmp     t_low,30000
        jne     set_flag
        and     t_low,0
        inc     t_high
        cmp     t_high,1440
        jne     set_flag
        and     t_high,0
set_flag:
        or      f,00000001b
akhir_int:
        sti
        iret
nmi    endp

```

```

;*****

```

```

;Procedures

```

```

;*****

```

```

;this procedure separate word result to 2 sending byte

```

```

;word mentioned is sysp, diasp, map, hr

```

```

separate    proc near

```

```

push        ax
        mov     ax,diasp
        cmp     ax,200
        jl      down1
        sub     ax,200
        mov     diaspt1,1
        mov     diaspt2,al
        jmp     spt1

```

```

down1:
        mov     diaspt1,0
        mov     diaspt2,al

```

```

spt1:
        mov     ax,sysp
        cmp     ax,200
        jl      down2
        sub     ax,200
        mov     syspt1,1
        mov     syspt2,al
        jmp     spt2

```

```

down2:
        mov     syspt1,0
        mov     syspt2,al

```

```

spt2:
        mov     ax,map
        cmp     ax,200
        jl      down3
        sub     ax,200
        mov     mappt1,1
        mov     mappt2,al
        jmp     spt3

```

```

down3:
        mov     mappt1,0

```

```

spt3:    mov     mapt2,al
        mov     ax,hr
        cmp     ax,200
        jl      down4
        sub     ax,200
        mov     hrt1,1
        mov     hrt2,al
        jmp     spt4
down4:   mov     hrt1,0
        mov     hrt2,al
spt4:   pop     ax
        ret
separate      endp

```

```

;=====
;blood pressure monitor
;=====
monitor proc near
    and     monfail,0
    mov     try,3
again:   call    setpressure
        call    delay
;setting initial condition
        dec     try
        mov     ax,t_low
        mov     oldtime,ax
        mov     totalpulse,0
        mov     first,1
        mov     tpduration,0
;observe for beginning pulse
pulsecheck1:
;expose cuff pressure
        call    get_data
        call    convert
        mov     cx,presure
        call    expose2
        mov     dx,port2_c
        in      al,dx
        and     al,100000000b
        cmp     al,00000000b
        jne     progress1
;gettime for newtime and see beginning delay
        mov     ax,t_low
        mov     newtime,ax
        cmp     ax,oldtime
        jl      abcount1
        sub     ax,oldtime
        mov     t,ax
        jmp     dcs1
abcount1:
        mov     bx,30000

```

```

        sub     bx,oldtime
        add     ax,bx
        mov     t,ax
dcs1:   mov     ax,7500 ;setting to 15 s beginning delay
        cmp     ax,t
        jl      try_again
        jmp     pulsecheck1
;try to monitor again ( try max twice attempt )
try_again:
        call    fail_sound
        cmp     try,0
        jne     again
        jmp     error1
progress1:
        mov     ax,t_low
        mov     newtime,ax
more:   call    get_data
        call    pulse_sound
        call    convert
        mov     cx,pressure
        call    expose3
;first pressure detect is systolic
        cmp     first,0
        jne     firstdetect
        mov     ax,freq
        mov     fdias,ax
        jmp     pulsecheck2
firstdetect:
        mov     first,0
        mov     ax,freq
        mov     fsys,ax
pulsecheck2:
;expose cuff pressure
        call    get_data
        call    convert
        mov     cx,pressure
        call    expose2
        mov     dx,port2_c
        in      al,dx
        and     al,10000000b
        cmp     al,10000000b
        je      pulsecheck2
        mov     ax,newtime
        mov     oldtime,ax
pulsecheck3:
;expose cuff pressure
        call    get_data
        call    convert
        mov     cx,pressure
        call    expose2
        mov     dx,port2_c
        in      al,dx
        and     al,10000000b

```

```

        cmp     al,00000000b
        jne     progress2
        mov     ax,t_low
        mov     newtime,ax
        cmp     ax,oldtime
        jl      abcount2
        sub     ax,oldtime
        mov     t,ax
        jmp     dcs2
abcount2:
        mov     bx,30000
        sub     bx,oldtime
        add     ax,bx
        mov     t,ax
dcs2:
        mov     ax,1000 ;setting to 2 s delay pulse max
        cmp     ax,t
        jl      endmonit
        jmp     pulsecheck3
progress2:
        mov     ax,t_low
        mov     newtime,ax
;count T heart rate
        mov     ax,newtime
        cmp     ax,oldtime
        jl      abcount3
        sub     ax,oldtime
        mov     pduration,ax
        jmp     progress3
abcount3:
        mov     bx,30000
        sub     bx,oldtime
        add     ax,bx
        mov     pduration,ax
progress3:
        inc     totalpulse
        mov     ax,tpduration
        add     ax,pduration
        mov     tpduration,ax
        jmp     more
endmonit:
;data valid if totalpulse greater equal 4
        cmp     totalpulse,4
        jge     bypass
        call    fail_sound
        cmp     try,0
        je      error1
        jmp     again
bypass:
;counting sys,dias,map,hr, then expose to display
        mov     ax,fsys
        mov     freq,ax
        call    convert
        mov     cx,pressure
        mov     sysp,cx

```



```

;expose systolic pressure
    call    expose3
    mov     ax,fdias
    mov     freq,ax
    call    convert
    mov     cx,pressure
    mov     diasp,cx
;expose diastolic pressure
    call    expose2
;map = diasp + ( sysp - diasp ) / 3
    mov     ax,sysp
    sub     ax,diasp
    mov     bl,3
    div     bl
    cmp     ah,5
    jg      increasel
    xor     ah,ah
    jmp     progress4
increasel:
    add     al,1
    xor     ah,ah
progress4:
    add     ax,diasp
    mov     cx,ax
    mov     map,ax
;expose mean artery pressure
    call    exposel
;count heart rate ( pulse / minutes )
    xor     dx,dx
    mov     ax,tpduration
    mov     bl,totalpulse
    xor     bh,bh
    div     bx
    mov     bx,ax
    mov     ax,30000
    xor     dx,dx
    div     bx
    mov     cx,ax
    mov     hr,ax
;expose heart rate
    call    expose4
    call    openv
    jmp     exitm
error1:
;if monitor fail old data expose
    call    openv
    mov     monfail,1
    mov     cx,sysp
    call    expose3
    mov     cx,diasp
    call    expose2
    mov     cx,map
    call    exposel
    mov     cx,hr
    call    expose4

```

```

        lea    bx,monitorf
        mov    expression,bx
        call   diagnose
        call   fail_sound
exitm:   ret
monitor endp
;*****
;compare and take decision
decision    proc near
;match sex
        cmp    dsex,0
        je     woman1
man:
;age compare
        cmp    dage,24
        jg     x1
        mov    dias_low,62
        mov    dias_high,88
        mov    sys_low,114
        mov    sys_high,142
        jmp    dagain1
;kangaroo help
woman1:
        jmp    woman
x1:
        cmp    dage,29
        jg     x2
        mov    dias_low,63
        mov    dias_high,87
        mov    sys_low,114
        mov    sys_high,142
        jmp    dagain1
x2:
        cmp    dage,34
        jg     x3
        mov    dias_low,63
        mov    dias_high,91
        mov    sys_low,113
        mov    sys_high,145
        jmp    dagain1
x3:
        cmp    dage,39
        jg     x4
        mov    dias_low,64
        mov    dias_high,94
        mov    sys_low,112
        mov    sys_high,148
        jmp    dagain1
x4:
        cmp    dage,44
        jg     x5
        mov    dias_low,67
        mov    dias_high,95
        mov    sys_low,113

```

```

        mov     sys_high,151
        jmp     dagainl
x5:     cmp     dage,49
        jg      x6
        mov     dias_low,69
        mov     dias_high,97
        mov     sys_low,114
        mov     sys_high,158
        jmp     dagainl
x6:     cmp     dage,54
        jg      x7
        mov     dias_low,71
        mov     dias_high,103
        mov     sys_low,118
        mov     sys_high,170
        jmp     dagainl
x7:     cmp     dage,59
        jg      x8
        mov     dias_low,72
        mov     dias_high,104
        mov     sys_low,123
        mov     sys_high,177
        jmp     dagainl
x8:     cmp     dage,64
        jg      x9
        mov     dias_low,75
        mov     dias_high,107
        mov     sys_low,128
        mov     sys_high,184
        jmp     dagainl
x9:     cmp     dage,69
        jg      x10
        mov     dias_low,72
        mov     dias_high,106
        mov     sys_low,128
        mov     sys_high,188
        jmp     dagainl
x10:    mov     dias_low,72
        mov     dias_high,106
        mov     sys_low,133
        mov     sys_high,197
        jmp     dagainl
woman:  ;age compare
        cmp     dage,24
        jg      y1
        mov     dias_low,60
        mov     dias_high,84
        mov     sys_low,108

```

```

        mov     sys_high,134
        jmp     dagainl
y1:
        cmp     dage,29
        jg      y2
        mov     dias_low,61
        mov     dias_high,85
        mov     sys_low,107
        mov     sys_high,137
        jmp     dagainl
y2:
        cmp     dage,34
        jg      y3
        mov     dias_low,63
        mov     dias_high,87
        mov     sys_low,109
        mov     sys_high,139
        jmp     dagainl
y3:
        cmp     dage,39
        jg      y4
        mov     dias_low,65
        mov     dias_high,91
        mov     sys_low,110
        mov     sys_high,144
        jmp     dagainl
y4:
        cmp     dage,44
        jg      y5
        mov     dias_low,67
        mov     dias_high,93
        mov     sys_low,112
        mov     sys_high,152
        jmp     dagainl
y5:
        cmp     dage,49
        jg      y6
        mov     dias_low,68
        mov     dias_high,100
        mov     sys_low,114
        mov     sys_high,166
        jmp     dagainl
y6:
        cmp     dage,54
        jg      y7
        mov     dias_low,71
        mov     dias_high,101
        mov     sys_low,119
        mov     sys_high,175
        jmp     dagainl
y7:
        cmp     dage,59
        jg      y8
        mov     dias_low,72
        mov     dias_high,104

```

```

        mov     sys_low,122
        mov     sys_high,178
        jmp     dagain1
y8:
        cmp     dage,64
        jg      y9
        mov     dias_low,74
        mov     dias_high,106
        mov     sys_low,128
        mov     sys_high,188
        jmp     dagain1
y9:
        cmp     dage,69
        jg      y10
        mov     dias_low,76
        mov     dias_high,106
        mov     sys_low,136
        mov     sys_high,196
        jmp     dagain1
y10:
        mov     dias_low,75
        mov     dias_high,107
        mov     sys_low,140
        mov     sys_high,202
        jmp     dagain1
dagain1:
;comparation to set decflag
;reset decflag
        and     decflag,0
;diastolic
        mov     ax,diasp
        cmp     ax,dias_low
        jl      pdiaslow
        cmp     ax,dias_high
        jg      pdiashigh
;systolic
syscheck:
        mov     ax,sysp
        cmp     ax,sys_low
        jl      psyslow
        cmp     ax,sys_high
        jg      psyshigh
        jmp     dagain2
pdiaslow:
        or      decflag,10000000b
        jmp     syscheck
pdiashigh:
        or      decflag,01000000b
        jmp     syscheck
psyslow:
        or      decflag,00100000b
        jmp     dagain2
psyshigh:
        or      decflag,00010000b
dagain2:

```

```

;make decision
    cmp     decflag,0
    je      normalc1
jmp xagain
;kangaroo help
normalc1:
    jmp     normalc
xagain:
;abnormal condition
    lea     bx,abnorbp
    mov     expression,bx
    call    diagnose
    mov     al,decflag
    and     al,10000000b
    cmp     al,10000000b
    jne     dagain3
    lea     bx,diaslow
    mov     expression,bx
    call    diagnose
dagain3:
    mov     al,decflag
    and     al,01000000b
    cmp     al,01000000b
    jne     dagain4
    lea     bx,diashigh
    mov     expression,bx
    call    diagnose
dagain4:
    mov     al,decflag
    and     al,00100000b
    cmp     al,00100000b
    jne     dagain5
    lea     bx,syslow
    mov     expression,bx
    call    diagnose
dagain5:
    mov     al,decflag
    and     al,00010000b
    cmp     al,00010000b
    jne     dagain6
    lea     bx,syshigh
    mov     expression,bx
    call    diagnose
dagain6:
;alarm ?
    cmp     dalarm,1
    jne     dagain7
;abnormal alarm
    call    start_sound
    call    fail_sound
dagain7:
    jmp     enddec
;normal condition
normalc:

```

```

        lea     bx,normalbp
        mov     expression,bx
        call    diagnose
enddec:  ret
decision endp

;*****
;data input
data_input proc near
dbegin:
;clear data
        mov     dmode,2 ;dmode = 0, manual; =1, automatic
        mov     dage,0
        mov     dsex,0
        mov     dperiode,0
        mov     dcontinuity,0
        mov     dalarm,0
        mov     ddiagnose,0

modein: lea     bx,mode
        mov     expression,bx
        mov     keypressed,0
        call    word_expose ;word_expose already call keyb_scan
        cmp     key,'A'
        je      automatic
        cmp     key,'B'
        je      manual
        jmp     modein
;manual mode
manual:
        mov     bx,expression
        add     bx,14
        mov     byte ptr [bx],24h
        add     bx,1
        mov     byte ptr [bx],0fh
        mov     keypressed,0
        call    word_expose
        cmp     key,'C'
        je      modeinh
        cmp     key,'E'
        je      gomanual
        jmp     manual
;kangaroo help
modeinh: jmp modein
gomanual:
        mov     dmode,0
        call    initialpress

acc:
        lea     bx,ok
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        cmp     key,'E'
        je      gomanual

```

```

        cmp     key, 'F'
        je      accl
        jmp     acc
accl:    jmp     data_end
;automatic mode
automatic:
        mov     dmode, 1
        mov     bx, expression
        add     bx, 14
        mov     byte ptr [bx], 0e0h
        add     bx, 1
        mov     byte ptr [bx], 0bh
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'C'
        je      modeinh
        cmp     key, 'E'
        je      initpr
        jmp     automatic
;initialize pressure charge to cuff at first time
initpr:
        call    initialpress
;inputting patient ages ( 0 - 99 years old )
agein:
        lea     bx, age
        mov     expression, bx
        mov     keypressed, 0
        call    word_expose
;age input
        cmp     key, 'A'
        je      agein
        cmp     key, 'B'
        je      agein
        cmp     key, 'C'
        je      agein
        cmp     key, 'D'
        je      agein
        cmp     key, 'E'
        je      agein
        cmp     key, 'F'
        je      agein
        sub     key, 30h
        mov     bl, key
        mov     al, 10
        mul     bl
        mov     dage, al
agein1:
        mov     bx, expression
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'A'
        je      agein1
        cmp     key, 'B'
        je      agein1

```



```

        cmp     key, 'C'
        je      agein1
        cmp     key, 'D'
        je      agein1
        cmp     key, 'E'
        je      agein1
        cmp     key, 'F'
        je      agein1
        sub     key, 30h
        mov     al, key
        add     dage, al
        xor     ch, ch
        mov     cl, dage
        call    expose3
agein2:
        mov     bx, expression
        mov     keypressed, 0
        call    word_expose
;sex determination input
        cmp     key, 'E'
        je      sexin
        cmp     key, 'C'
        je      ageinh
        jmp     agein2
;kangaroo help
ageinh: jmp agein
sexin:
        call    dreset
        lea     bx, sex
        mov     expression, bx
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'A'
        je      male
        cmp     key, 'B'
        je      female
        jmp     sexin
male:
        mov     dsex, 1
        mov     bx, expression
        add     bx, 14
        mov     byte ptr [bx], 24h
        add     bx, 1
        mov     byte ptr [bx], 0fh
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'C'
        je      sexin
        cmp     key, 'E'
        je      periodein
        jmp     male
female:
        mov     dsex, 0
        mov     bx, expression
        add     bx, 14

```

```

        mov     byte ptr [bx],0f8h
        add     bx,1
        mov     byte ptr [bx],0bh
        mov     keypressed,0
        call    word expose
        cmp     key,'C'
        je     sexin
        cmp     key,'E'
        je     periodein
        jmp     female
;perioda input, time duration in minutes ( 0 - 69 minutes )
periodein:
        lea     bx,periode
        mov     expression,bx
        mov     keypressed,0
        call    word expose
        cmp     key,'0'
        je     nul2
        cmp     key,'1'
        je     one2
        cmp     key,'2'
        je     two2
        cmp     key,'3'
        je     three2
        cmp     key,'4'
        je     four2
        cmp     key,'5'
        je     five2
        cmp     key,'6'
        je     six2
        jmp     periodein
nul2:   mov     dperiode,0
        jmp     dprogress3
one2:   mov     dperiode,10
        jmp     dprogress3
two2:   mov     dperiode,20
        jmp     dprogress3
three2: mov     dperiode,30
        jmp     dprogress3
four2:  mov     dperiode,40
        jmp     dprogress3
five2:  mov     dperiode,50
        jmp     dprogress3
six2:   mov     dperiode,60
dprogress3:
        mov     bx,expression
        mov     keypressed,0
        call    word expose
        cmp     key,'A'
        je     dprogress3
        cmp     key,'B'
        je     dprogress3
        cmp     key,'C'
        je     dprogress3
        cmp     key,'D'

```

```

        je      dprogress3
        cmp     key,'E'
        je      dprogress3
        cmp     key,'F'
        je      dprogress3
        sub     key,30h
        mov     al,key
        xor     ah,ah
        add     dperiode,ax
        mov     cx,dperiode
        call    expose3
dprogress4:
        mov     bx,expression
        mov     keypressed,0
        call    word_expose
;continuity input, time duration in hour ( 0 - 24 hours )
        cmp     key,'E'
        je      continuityin
        cmp     key,'C'
        je      periodeinh
        jmp     dprogress4
;kangaroo help
periodeinh:
        jmp     periodein
continuityin:
        call    dreset
        lea     bx,continuity
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        cmp     key,'0'
        je      nul4
        cmp     key,'1'
        je      one4
        cmp     key,'2'
        je      two4
        jmp     continuityin
nul4:    mov     dcontinuity,0
        jmp     dprogress5
one4:    mov     dcontinuity,10
        jmp     dprogress5
two4:    mov     dcontinuity,20
dprogress5:
        mov     bx,expression
        mov     keypressed,0
        call    word_expose
        cmp     key,'A'
        je      dprogress5
        cmp     key,'B'
        je      dprogress5
        cmp     key,'C'
        je      dprogress5
        cmp     key,'D'
        je      dprogress5
        cmp     key,'E'

```

```

        je      dprogress5
        cmp     key,'F'
        je      dprogress5
        sub     key,30h
        mov     al,key
        xor     ah,ah
        add     dcontinuity,ax
        mov     cx,dcontinuity
        call    expose3
continuity1:
        mov     bx,expression
        mov     keypressed,0
        call    word_expose
;alarm extension if abnormal condition occur
        cmp     key,'E'
        je      alarmin
        cmp     key,'C'
        je      continuityinh
        jmp     continuity1
;kangaroo help
continuityinh:
        jmp     continuityin
alarmin:
        call    dreset
        lea     bx,alarm
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        cmp     key,'A'
        je      yes1
        cmp     key,'B'
        je      nol
        jmp     alarmin
yes1:
        mov     dalarm,1
        mov     bx,expression
        add     bx,14
        mov     byte ptr [bx],0c5h
        add     bx,1
        mov     byte ptr [bx],0bh
        mov     keypressed,0
        call    word_expose
        cmp     key,'C'
        je      alarmin
        cmp     key,'E'
        je      diagnin
        jmp     yes1
nol:
        mov     dalarm,0
        mov     bx,expression
        add     bx,14
        mov     byte ptr [bx],0a4h
        add     bx,1
        mov     byte ptr [bx],0eh

```

```

        mov     keypressed,0
        call    word_expose
        cmp     key,'C'
        je      alarmin
        cmp     key,'E'
        je      diagnin
        jmp     nol
;diagnose after monitoring ?
diagnin:
        lea     bx,diagns
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        cmp     key,'A'
        je      yes2
        cmp     key,'B'
        je      no2
        jmp     diagnin
yes2:
        mov     ddiagnose,1
        mov     bx,expression
        add     bx,14
        mov     byte ptr [bx],0c5h
        add     bx,1
        mov     byte ptr [bx],0bh
        mov     keypressed,0
        call    word_expose
        cmp     key,'C'
        je      diagnin
        cmp     key,'E'
        je      comlinkin
        jmp     yes2
no2:
        mov     ddiagnose,0
        mov     bx,expression
        add     bx,14
        mov     byte ptr [bx],0a4h
        add     bx,1
        mov     byte ptr [bx],0eh
        mov     keypressed,0
        call    word_expose
        cmp     key,'C'
        je      diagnin
        cmp     key,'E'
        je      comlinkin
        jmp     no2
;kangaroo help
diagninh:
        jmp     diagnin
;diagnose after monitoring ?
comlinkin:
        lea     bx,comlink
        mov     expression,bx
        mov     keypressed,0
        call    word_expose

```

```

        cmp     key, 'A'
        je      yes3
        cmp     key, 'B'
        je      no3
        jmp     comlinkin
no3:
        mov     dcomlink, 0
        mov     bx, expression
        add     bx, 14
        mov     byte ptr [bx], 0a4h
        add     bx, 1
        mov     byte ptr [bx], 0eh
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'C'
        je      comlinkin
        cmp     key, 'E'
        je      okin
        jmp     no3
yes3:
        mov     dcomlink, 1
        lea     bx, comlink
        mov     expression, bx
        add     bx, 14
        mov     byte ptr [bx], 0c5h
        add     bx, 1
        mov     byte ptr [bx], 0bh
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'C'
        je      comlinkin
        cmp     key, 'D'
        je      ctest
        cmp     key, 'E'
        je      okin
        jmp     yes3
ctest:
        lea     bx, com_test
        mov     expression, bx
        mov     keypressed, 0
        call    word_expose
        cmp     key, 'F'
        je      do_test
        cmp     key, 'C'
        je      yes3
        jmp     ctest
do_test:
        call    comtest
        jmp     yes3

;kangaroo help
automatic:
        lea     bx, mode
        mov     expression, bx
        jmp     automatic

```

```

;data ok ?
okin:      lea      bx,ok
           mov      expression,bx
           mov      keypressed,0
           call     word_expose
           cmp      key,'E'
           je       automatic
           cmp      key,'F'
           je       data_end
           jmp      okin
data_end:  ret
data_input      endp

;*****
;set initial pressure to inflate cuff max 300 mmHg
initialpress    proc near
push    cx
;set pressure 150 at first entry
        mov      dpressure,150
        mov      cx,dpressure
        call     expose3
pressurein:
        lea      bx,pset
        mov      expression,bx
        mov      keypressed,0
        call     word_expose
;set select initial pressure by dec / inc 10
        cmp      key,'A'
        je       incl0
        cmp      key,'B'
        je       decl0
        cmp      key,'E'
        je       endpresin
        jmp      pressurein
incl0:
        cmp      dpressure,300
        je       fix300
        add      dpressure,10
fix300:
        mov      cx,dpressure
        call     expose3
        jmp      pressurein
decl0:
        cmp      dpressure,150
        je       fix150
        sub      dpressure,10
fix150:
        mov      cx,dpressure
        call     expose3
        jmp      pressurein
endpresin:
;reset numerik display
        call     dreset

```

```

pop      cx
ret
initialpress  endp

;*****
;setting cuff pressure
setpressure  proc near
    jmp      acstart
;help for far jumping
set280j:
    jmp      set280
set290j:
    jmp      set290
acstart:
;find aircharge value range
    mov      ax,syssp
    add      ax,30
    cmp      ax,150
    jle      set150
    cmp      ax,160
    jle      set160
    cmp      ax,170
    jle      set170
    cmp      ax,180
    jle      set180
    cmp      ax,190
    jle      set190
    cmp      ax,200
    jle      set200
    cmp      ax,210
    jle      set210
    cmp      ax,220
    jle      set220
    cmp      ax,230
    jle      set230
    cmp      ax,240
    jle      set240
    cmp      ax,250
    jle      set250
    cmp      ax,260
    jle      set260
    cmp      ax,270
    jle      set270
    cmp      ax,280
    jle      set280j
    cmp      ax,290
    jle      set290j
;setting aircharge value
    mov      ax,p300
    mov      aircharge,ax
    jmp      acprogress
set150: mov      ax,p150
    mov      aircharge,ax
    jmp      acprogress
set160: mov      ax,p160

```



```

        mov     aircharge,ax
        jmp     acprogress
set170: mov     ax,p170
        mov     aircharge,ax
        jmp     acprogress
set180: mov     ax,p180
        mov     aircharge,ax
        jmp     acprogress
set190: mov     ax,p190
        mov     aircharge,ax
        jmp     acprogress
set200: mov     ax,p200
        mov     aircharge,ax
        jmp     acprogress
set210: mov     ax,p210
        mov     aircharge,ax
        jmp     acprogress
set220: mov     ax,p220
        mov     aircharge,ax
        jmp     acprogress
set230: mov     ax,p230
        mov     aircharge,ax
        jmp     acprogress
set240: mov     ax,p240
        mov     aircharge,ax
        jmp     acprogress
set250: mov     ax,p250
        mov     aircharge,ax
        jmp     acprogress
set260: mov     ax,p260
        mov     aircharge,ax
        jmp     acprogress
set270: mov     ax,p270
        mov     aircharge,ax
        jmp     acprogress
set280: mov     ax,p280
        mov     aircharge,ax
        jmp     acprogress
set290: mov     ax,p290
        mov     aircharge,ax
        jmp     acprogress
acprogress:
;open air-charge to cuff, close release valve
        mov     dx,port1_b
        mov     al,11111111b
        out     dx,al
checkp:
        call    get_data
        mov     ax,aircharge
        cmp     freq,ax
        jle     pressmatch
;expose cuff pressure
        call    convert
        mov     cx,pressure
        call    expose2

```

```

        jmp      checkp
pressmatch:
        call     convert
        mov      cx,pressure
        call     expose3
;close air-charge to cuff
        mov      dx,port1_b
        mov      al,1111110b
        out      dx,al
        ret
setpressure      endp

;*****
;release cuff pressure
openv  proc near
push ax
push dx
        mov      dx,port1_b
        mov      al,11111100b      ;pb1 = 0 valve is open
        out      dx,al
pop dx
pop ax
        ret
openv  endp

;*****
;time delay
delay  proc near
push cx
        mov      cx,0ffffh
ulangi: loop     ulangi
pop cx
        ret
delay  endp

;*****
;getting pulsa/10ms
get_data proc near
;count pulses in 10 ms / 5x NMI
        push     cx
        and      cx,0
        and      f,0
int_cek:
        and      f,00000001b
        jz       int_cek
;counter 0, mode 0, binary counter, initial value is 9999
;or 270fh
        mov      dx,counter0
        mov      al,0fh
        out      dx,al
        mov      al,27h
        out      dx,al
;flag int check recover
        and      f,0
;wait for 5x NMI

```

```

wait_5nmi:
not_yet:
    and    f,00000001b
    jz     not_yet
    inc    cx
    and    f,0
    cmp    cx,5
    je     counter_latch
    jmp    wait_5nmi
counter_latch:
;latching
    mov    al,00000000b
    mov    dx,cwordc
    out    dx,al
    mov    dx,counter0
    in     al,dx
    mov    bl,al
    in     al,dx
    mov    bh,al
    mov    ax,10000
    sub    ax,bx
    mov    freq,ax
    pop    cx
    ret
get_data endp

;*****
;reset numerik display
;1-4
dreset proc near
push ax
push dx
    mov    dx,port1_a
;reset display
    mov    al,01010101b
    out    dx,al
    mov    al,00000000b
    out    dx,al
pop dx
pop ax
    ret
dreset endp

;*****
; numeric display expose
; 1 - 4
exposel proc near
;program jalankan led
    mov    dx,port1_a
;reset old data
    mov    al,01000000b
    out    dx,al
count1:
hl:
    mov    al,10000000b

```

```

;sis counter di bx
;counter awal 9999+1
;hasil perhitungan pulsa selama
;10ms = ax

```

```

        out      dx,al
11:      mov      al,00000000b
        out      dx,al
loop     count1
        ret
expose1 endp

expose2 proc near
;program jalankan led
        mov      dx,port1_a
;reset old data
        mov      al,00010000b
        out      dx,al
count2:
h2:      mov      al,00100000b
        out      dx,al
12:      mov      al,00000000b
        out      dx,al
loop     count2
        ret
expose2 endp

expose3 proc near
;program jalankan led
        mov      dx,port1_a
;reset old data
        mov      al,00000100b
        out      dx,al
count3:
h3:      mov      al,00001000b
        out      dx,al
13:      mov      al,00000000b
        out      dx,al
loop     count3
        ret
expose3 endp

expose4 proc near
;program jalankan led
        mov      dx,port1_a
;reset old data
        mov      al,00000001b
        out      dx,al
count4:
h4:      mov      al,00000010b
        out      dx,al
14:      mov      al,00000000b
        out      dx,al

```

```

loop    count4
ret
expose4 endp

```

```

;*****
;frequency to pressure conversion
;with interpolation between two range point, which frequency value
;to convert ( to pressure ) in that range.
;interpolation equation is  $P = - (p1-p2) * (F-f1) / (f2-f1) + p1$ 
;(f1,p1) dan (f2,p2) are standard point get from observation

```

```

convert proc near
;take frequency value to convert

```

```

        mov     ax,freq
        jmp     compare

```

```

;help for far jump

```

```

range1s:

```

```

        jmp     range1

```

```

range2s:

```

```

        jmp     range2

```

```

range3s:

```

```

        jmp     range3

```

```

range4s:

```

```

        jmp     range4

```

```

range5s:

```

```

        jmp     range5

```

```

range6s:

```

```

        jmp     range6

```

```

range7s:

```

```

        jmp     range7

```

```

range8s:

```

```

        jmp     range8

```

```

range9s:

```

```

        jmp     range9

```

```

range10s:

```

```

        jmp     range10

```

```

range11s:

```

```

        jmp     range11

```

```

range12s:

```

```

        jmp     range12

```

```

range13s:

```

```

        jmp     range13

```

```

range14s:

```

```

        jmp     range14

```

```

range15s:

```

```

        jmp     range15

```

```

range16s:

```

```

        jmp     range16

```

```

range17s:

```

```

        jmp     range17

```

```

range18s:

```

```

        jmp     range18

```

```

range19s:

```

```

        jmp     range19

```

```

range20s:

```

```

        jmp      range20
range21s:
        jmp      range21
range22s:
        jmp      range22
range23s:
        jmp      range23
range24s:
        jmp      range24
range25s:
        jmp      range25
;chose range frekuensi value
compare:
        cmp      ax,6491
        jge      range1s
        cmp      ax,6387
        jge      range2s
        cmp      ax,6287
        jge      range3s
        cmp      ax,6175
        jge      range4s
        cmp      ax,6070
        jge      range5s
        cmp      ax,5965
        jge      range6s
        cmp      ax,5863
        jge      range7s
        cmp      ax,5761
        jge      range8s
        cmp      ax,5660
        jge      range9s
        cmp      ax,5558
        jge      range10s
        cmp      ax,5458
        jge      range11s
        cmp      ax,5357
        jge      range12s
        cmp      ax,5258
        jge      range13s
        cmp      ax,5158
        jge      range14s
        cmp      ax,5059
        jge      range15s
        cmp      ax,4962
        jge      range16s
        cmp      ax,4861
        jge      range17s
        cmp      ax,4767
        jge      range18s
        cmp      ax,4668
        jge      range19s
        cmp      ax,4572
        jge      range20s
        cmp      ax,4474
        jge      range21s

```

```

        cmp     ax,4379
        jge     range22s
        cmp     ax,4282
        jge     range23s
        cmp     ax,4190
        jge     range24s
        cmp     ax,4096
        jge     range25s
        cmp     ax,4001
        jge     range26
        cmp     ax,3923
        jge     range27
        cmp     ax,3808
        jge     range28
        cmp     ax,3712
        jge     range29
;take value for (f1,p1) dan (f2,p2)
        mov     f1,3612
        mov     f2,3712
        mov     p1,300
        mov     p2,290
        jmp     equation
range29:
        mov     f1,3712
        mov     f2,3808
        mov     p1,290
        mov     p2,280
        jmp     equation
range28:
        mov     f1,3808
        mov     f2,3923
        mov     p1,280
        mov     p2,270
        jmp     equation
range27:
        mov     f1,3923
        mov     f2,4001
        mov     p1,270
        mov     p2,260
        jmp     equation
range26:
        mov     f1,4001
        mov     f2,4096
        mov     p1,260
        mov     p2,250
        jmp     equation
range25:
        mov     f1,4096
        mov     f2,4190
        mov     p1,250
        mov     p2,240
        jmp     equation
range24:
        mov     f1,4190
        mov     f2,4282

```

```

        mov     p1,240
        mov     p2,230
        jmp     equation
range23:
        mov     f1,4282
        mov     f2,4379
        mov     p1,230
        mov     p2,220
        jmp     equation
range22:
        mov     f1,4379
        mov     f2,4474
        mov     p1,220
        mov     p2,210
        jmp     equation
range21:
        mov     f1,4474
        mov     f2,4572
        mov     p1,210
        mov     p2,200
        jmp     equation
range20:
        mov     f1,4572
        mov     f2,4668
        mov     p1,200
        mov     p2,190
        jmp     equation
range19:
        mov     f1,4668
        mov     f2,4767
        mov     p1,190
        mov     p2,180
        jmp     equation
range18:
        mov     f1,4767
        mov     f2,4861
        mov     p1,180
        mov     p2,170
        jmp     equation
range17:
        mov     f1,4861
        mov     f2,4962
        mov     p1,170
        mov     p2,160
        jmp     equation
range16:
        mov     f1,4962
        mov     f2,5059
        mov     p1,160
        mov     p2,150
        jmp     equation
range15:
        mov     f1,5059
        mov     f2,5158
        mov     p1,150

```



```

        mov     p2,140
        jmp     equation
range14:
        mov     f1,5158
        mov     f2,5258
        mov     p1,140
        mov     p2,130
        jmp     equation
range13:
        mov     f1,5258
        mov     f2,5357
        mov     p1,130
        mov     p2,120
        jmp     equation
range12:
        mov     f1,5357
        mov     f2,5458
        mov     p1,120
        mov     p2,110
        jmp     equation
range11:
        mov     f1,5458
        mov     f2,5558
        mov     p1,110
        mov     p2,100
        jmp     equation
range10:
        mov     f1,5558
        mov     f2,5660
        mov     p1,100
        mov     p2,90
        jmp     equation
range9:
        mov     f1,5660
        mov     f2,5761
        mov     p1,90
        mov     p2,80
        jmp     equation
range8:
        mov     f1,5761
        mov     f2,5863
        mov     p1,80
        mov     p2,70
        jmp     equation
range7:
        mov     f1,5863
        mov     f2,5965
        mov     p1,70
        mov     p2,60
        jmp     equation
range6:
        mov     f1,5965
        mov     f2,6070
        mov     p1,60
        mov     p2,50

```

```

        jmp      equation
range5:  mov      f1,6070
        mov      f2,6175
        mov      p1,50
        mov      p2,40
        jmp      equation
range4:  mov      f1,6175
        mov      f2,6287
        mov      p1,40
        mov      p2,30
        jmp      equation
range3:  mov      f1,6287
        mov      f2,6387
        mov      p1,30
        mov      p2,20
        jmp      equation
range2:  mov      f1,6387
        mov      f2,6491
        mov      p1,20
        mov      p2,10
        jmp      equation
range1:  mov      f1,6491
        mov      f2,6608
        mov      p1,10
        mov      p2,0
        jmp      equation
;bentuk persamaan konversi
equation:
push ax
push bx
push cx
push dx
;ax = p1-p2
        mov      ax,p1
        sub      ax,p2
;bx = freq-f1
        mov      bx,freq
        sub      bx,f1
;cx = f2-f1
        mov      cx,f2
        sub      cx,f1
;multi ax dan bx
        mul      bx                ;result dx..ax
;result divided by cx
        mov      bx,cx
        div      bx                ;result ax, rest dx
;dx greater equal 5, thus ax = ax + 1
        cmp      dx,5
        jl       steady
        add      ax,1

```

```

steady:
;account result is added pl
    sub    pl,ax    ;conversion result at ax
    mov    ax,pl
    mov    pressure,ax
pop dx
pop cx
pop bx
pop ax
convert endp

;*****
keyb_scan    proc near
push ax
push bx
push cx
push dx
Begin_scan:  mov    cl,11111110b
             lea    bx,key_data
col_scan:    mov    al,cl
             mov    dx,port1_c    ;scan out
             out    dx,al
             mov    dx,port1_c    ;scan in
             in     al,dx
             and    al,11110000b
             cmp    al,11110000b
             jne    knob_pushed    ;jump to next-search
             rol    cl,1            ;programme
             add    bx,4
             cmp    cl,11101111b    ;inspect that all coloms
             jne    col_scan        ;have been scanned
             jmp    endscan
knob_pushed: mov    ah,0
             call   keyb_sound
             mov    keypressed,1
search:      shl    al,1
             jnc    knob_name
             inc    ah
             jmp    search
knob_name:   mov    al,ah
             xor    ah,ah
             add    bx,ax
             mov    al,byte ptr ds:[bx]    ;al is knob_name
             mov    key,al
endscan:
pop dx
pop cx
pop bx
pop ax
             ret
keyb_scan    endp

;*****
start_sound    proc near

```

```

; kirim nada do oktaf 5, f = 523Hz
    mov     dx, counter2
    mov     al, 24h
    out     dx, al
    mov     al, 18h
    out     dx, al
    mov     cx, 5
s0:      push    cx
    mov     cx, 30000
s00:     loop   s00
    pop     cx
    loop    s0
    mov     al, silent
    mov     dx, cwordc
    out     dx, al
    mov     al, 10110111b    ; counter2, word, mode 3, bcd counter
    mov     dx, cwordc
    out     dx, al
    ret
start_sound    endp

end_sound      proc near
; kirim nada A, oktaf 4, f = 440Hz, 5x
    mov     cx, 5
s1:      push    cx
    mov     dx, counter2
    mov     al, 68h
    out     dx, al
    mov     al, 21h
    out     dx, al
    mov     cx, 65000
s2:      loop   s2
    mov     al, silent
    mov     dx, cwordc
    out     dx, al
    mov     cx, 60000
s3:      loop   s3
    mov     al, 10110111b    ; counter2, word, mode 3, bcd counter
    mov     dx, cwordc
    out     dx, al
    pop     cx
    loop    s1
    ret
end_sound      endp

pulse_sound    proc near
; kirim nada do oktaf 8, f = 4186Hz
    mov     dx, counter2
    mov     al, 28h
    out     dx, al
    mov     al, 02h
    out     dx, al
    mov     cx, 65000
s4:      loop   s4
    mov     al, silent

```

```

        mov     dx,cwordc
        out     dx,al
        mov     al,10110111b    ;counter2, word, mode 3, bcd counter
        mov     dx,cwordc
        out     dx,al
        ret
pulse_sound     endp

keyb_sound     proc near
push ax
push bx
push cx
push dx
;kirim nada do oktaf 7, f = 2093Hz
        mov     dx,counter2
        mov     al,56h
        out     dx,al
        mov     al,05h
        out     dx,al
        mov     cx,65000
s5:      loop    s5
        mov     al,silent
        mov     dx,cwordc
        out     dx,al
        mov     al,10110111b    ;counter2, word, mode 3, bcd counter
        mov     dx,cwordc
        out     dx,al
pop dx
pop cx
pop bx
pop ax
        ret
keyb_sound     endp

fail_sound     proc near
;kirim nada C, oktaf 6, f = 1047Hz, 5x
        mov     cx,20
s6:      push    cx
        mov     dx,counter2
        mov     al,11h
        out     dx,al
        mov     al,09h
        out     dx,al
        mov     cx,25000
s7:      loop    s7
        mov     al,silent
        mov     dx,cwordc
        out     dx,al
        mov     cx,60000
s8:      loop    s8
        mov     al,10110111b    ;counter2, word, mode 3, bcd counter
        mov     dx,cwordc
        out     dx,al
        pop     cx
        loop    s6

```

```

        ret
fail_sound    endp

;*****
;display procedure
fdelay proc near
push cx
        mov     cx,750
label1: loop   label1
pop cx
        ret
fdelay endp
;
ndelay proc near
push cx
        mov     cx,60000
label2: loop   label2
pop cx
        ret
ndelay endp
;
word_expose proc near
push ax
push bx
push cx
push dx
        mov     geser,01111111b
        mov     bx,expression
        mov     reference,bx
repeat1:
        mov     bx,reference
        mov     cx,8
frame1:
        mov     al,byte ptr [bx]
        mov     pb,al
        mov     al,byte ptr [bx+1]
        mov     pc,al
;character multiplexing
        mov     dx,port2_a
        mov     al,geser
        out     dx,al
;one character expose
        mov     dx,port2_b
        mov     al,pb
        out     dx,al
        mov     dx,port2_c
        mov     al,pc
        out     dx,al
;delay time
        call    fdelay
;next position data
        ror     geser,1
;next data
        add     bx,2
;expose again

```



MILIK PERPUSTAKAAN
 INSTITUT TEKNOLOGI SEPULUH NOPEMBER
 SURABAYA

```

        loop    frame1
;look if the keypressed
        call    keyb_scan
        cmp     keypressed,1
        jne     repeat1

pop dx
pop cx
pop bx
pop ax

        ret
word_expose endp
;
waitexpose    proc near
push ax
push dx
;character position
        mov     dx,port2_a
        mov     al,11111110b
        out     dx,al
;character 'W' expose
        mov     dx,port2_b
        mov     al,0e4h
        out     dx,al
        mov     dx,port2_c
        mov     al,0ch
        out     dx,al

pop dx
pop ax

        ret
waitexpose    endp
;
autoexpose    proc near
push ax
push dx
;character position
        mov     dx,port2_a
        mov     al,11111110b
        out     dx,al
;character 'A' expose
        mov     dx,port2_b
        mov     al,0e0h
        out     dx,al
        mov     dx,port2_c
        mov     al,0bh
        out     dx,al

pop dx
pop ax

        ret
autoexpose    endp
;
manualexpose    proc near
push ax
push dx
;character position
        mov     dx,port2_a

```

```

        mov     al,11111110b
        out     dx,al
;character 'M' expose
        mov     dx,port2_b
        mov     al,24h
        out     dx,al
        mov     dx,port2_c
        mov     al,0fh
        out     dx,al
pop dx
pop ax
        ret
manualexpose     endp
;
diagnose proc near
        mov     geser,01111111b
        mov     bx,expression
        mov     reference,bx
repeat2:
        mov     frametime,10
tframe:
        mov     cx,8
frame2:
        mov     al,byte ptr [bx]
        mov     pb,al
        mov     al,byte ptr [bx+1]
        mov     pc,al
;character multiplexing
        mov     dx,port2_a
        mov     al,geser
        out     dx,al
;one character expose
        mov     dx,port2_b
        mov     al,pb
        out     dx,al
        mov     dx,port2_c
        mov     al,pc
        out     dx,al
;delay time for one character
        call    fdelay
;next position data
        ror     geser,1
;next data to 8 digit shift right
        add     bx,2
;expose again
        loop    frame2
;one frame delay time
        mov     bx,reference
        sub     frametime,1
        cmp     frametime,0
        jne     tframe
;display blank to expose new frame
        mov     al,0ffh
        mov     dx,port2_a
        out     dx,al

```



```

;new data, untill 00h
    add     reference,2
    mov     bx,reference
;last data ?
    cmp     byte ptr [bx+15],00h
    je      exit1
    jmp     repeat2
exit1:
    ret
diagnose endp

;*****
;usart procedure
;
usart_out     proc near
push    ax
push    bx
push    cx
push    dx
    mov     dx,U_control_add
    mov     cx,65000

test1:
;if fail try 65000 x
    dec     cx
    cmp     cx,0
    je      fail_out
    in      al,dx
    and     al,10000001b    ;data set ready and transmitter ready
?
    cmp     al,10000001b
    jne     test1
    mov     dx,U_data_add
    mov     al,transmited
    out     dx,al
    mov     failflag,0
    jmp     end_out
fail_out:
    mov     failflag,1
    lea     bx,fail
    mov     expression,bx
    mov     keypressed,0
    call    word_expose

end_out:
pop     dx
pop     cx
pop     bx
pop     ax
    ret
usart_out     endp

usart_in     proc near
push    ax
push    bx
push    cx
push    dx

```

```

        mov     dx,U_control_add
        mov     cx,65000
test2:
;if fail try 65000 x
        dec     cx
        cmp     cx,0
        je      fail_in
        in      al,dx
        and     al,00000010b    ;receiver ready ?
        cmp     al,00000000b
        je      test2
        mov     dx,U_data_add
        in      al,dx
        mov     received,al
        mov     failflag,0
        jmp     end_in
fail_in:
        mov     failflag,1
        lea     bx,fail
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
end_in:
        pop     dx
        pop     cx
        pop     bx
        pop     ax
        ret
usart_in      endp

synchron      proc near
push ax
sync:  mov     transmited,'$'
        call    usart_out
        call    usart_in
        mov     al,received
        cmp     al,'#'
        jne     sync
pop ax
        ret
synchron      endp

comtest proc near
        mov     transmited,'$'
        call    usart_out
        cmp     failflag,1
        je      endcomtest
        call    usart_in
        cmp     failflag,1
        je      endcomtest
        mov     al,received
        cmp     al,'#'
        jne     link_fail
        mov     transmited,'$'
        call    usart_out

```

```

        cmp     failflag,1
        je      endcomtest
        lea     bx,pass
        mov     expression,bx
        mov     keypressed,0
        call    word_expose
        jmp     endcomtest

```

```

link_fail:
        lea     bx,fail
        mov     expression,bx
        mov     keypressed,0
        call    word_expose

```

```

endcomtest:
        ret

```

```

comtest endp

```

```

;*****

```

```

;5. Tempat address program jmp / 0EAh

```

```

yy  db (xx-yy) dup(0ffh)      ; ISI DG OFFH SEBANYAK (XX-YY)
org 1ff0h                    ; SET KE ENTRY POINT RESET 0FFFF0H

```

```

xx  db 0eah                  ; JMP SA:ORG

```

```

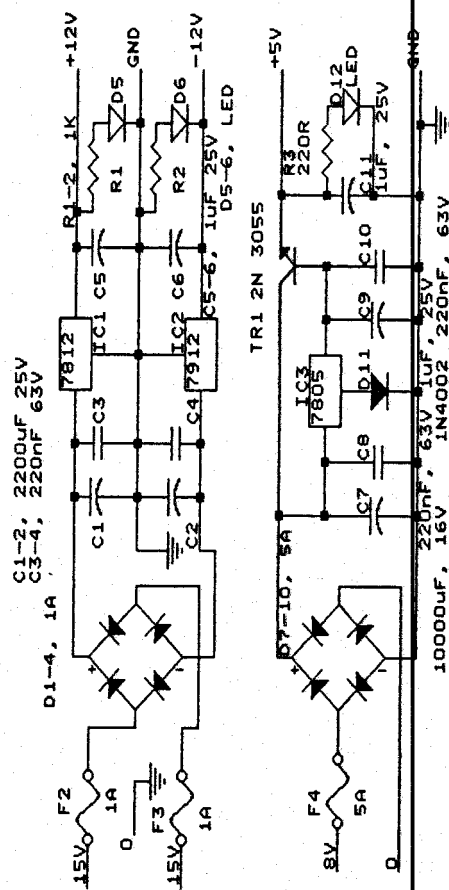
        dw offset begin,0fe00h
        db 11 dup(0ffh)

```

```

; ISI DG OFFH SEBANYAK 11

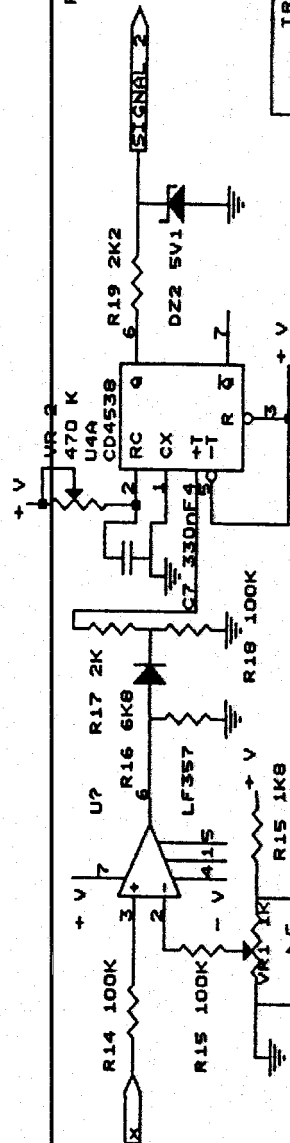
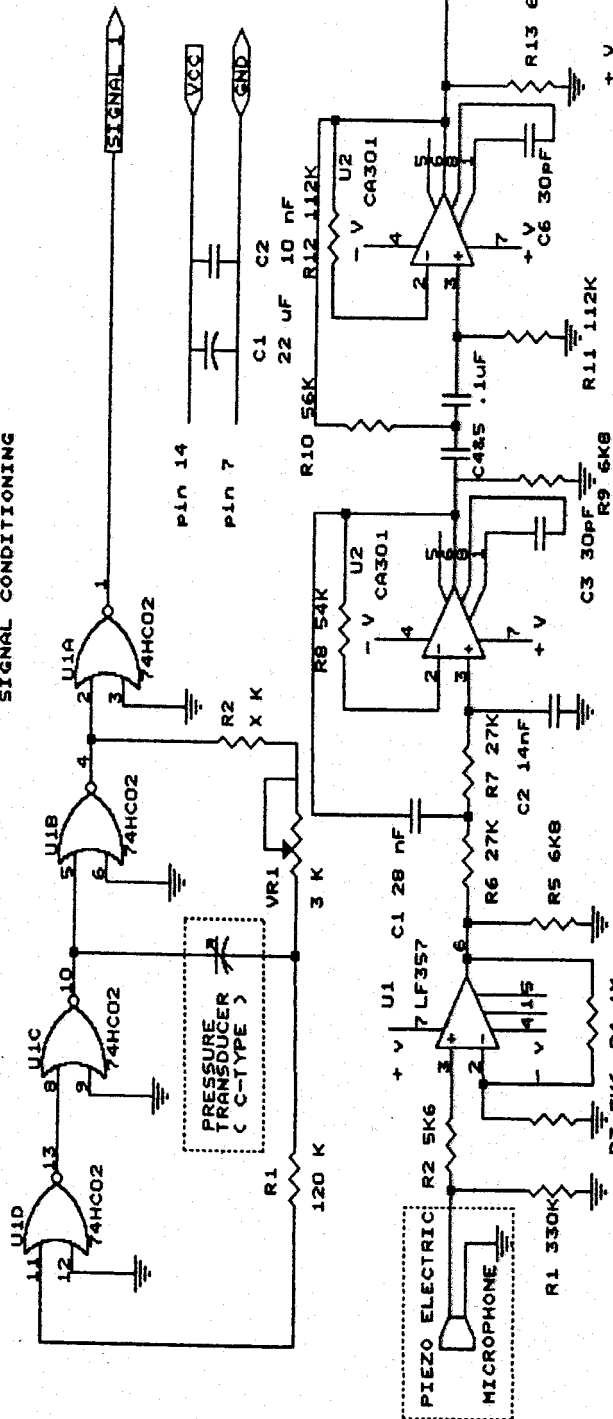
```



Title		PSU UNIT
Size		Document Number
A		
Date:	January 1, 1980	Sheet of
REV		

PRESSURE TRANSDUCER

SIGNAL CONDITIONING

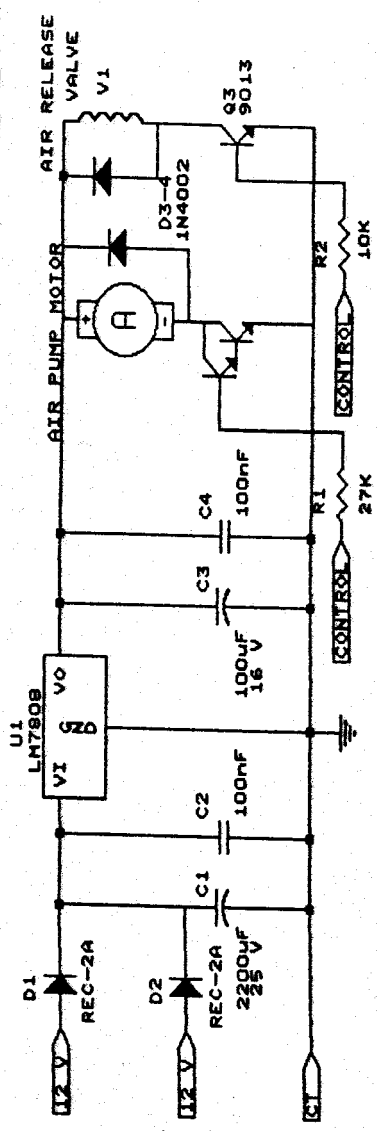


TRANSDUCER & SIGNAL CONDITIONING		
Size	Document Number	REV
A	ONE	AS
Date:	January 1, 1980	Sheet of

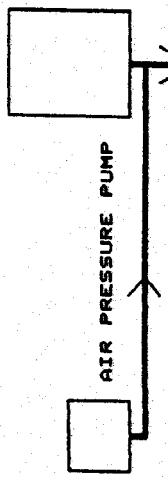
SIGNAL CONDITIONING

KOROTKOFF SOUND TRANSDUCER

BLOOD PRESSURE MONITOR PNEUMATIC SYSTEM

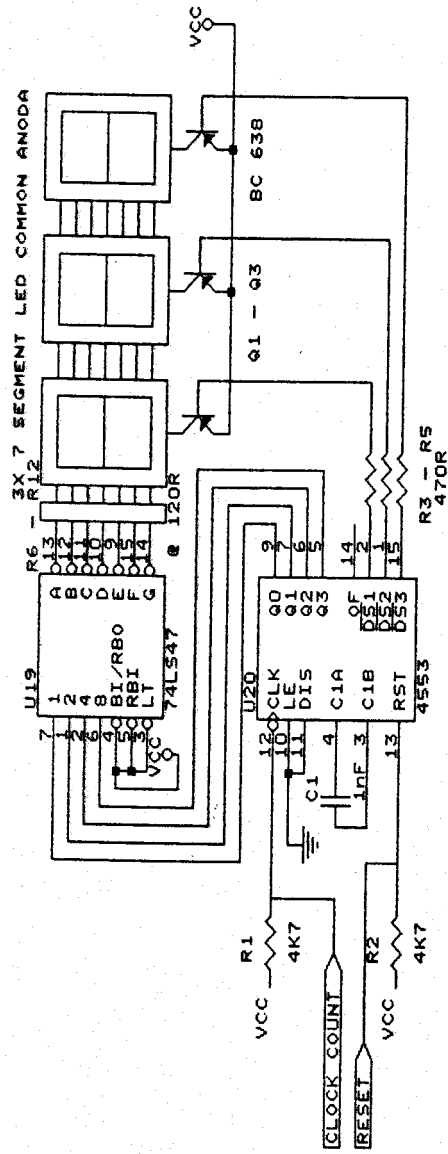


CUFF

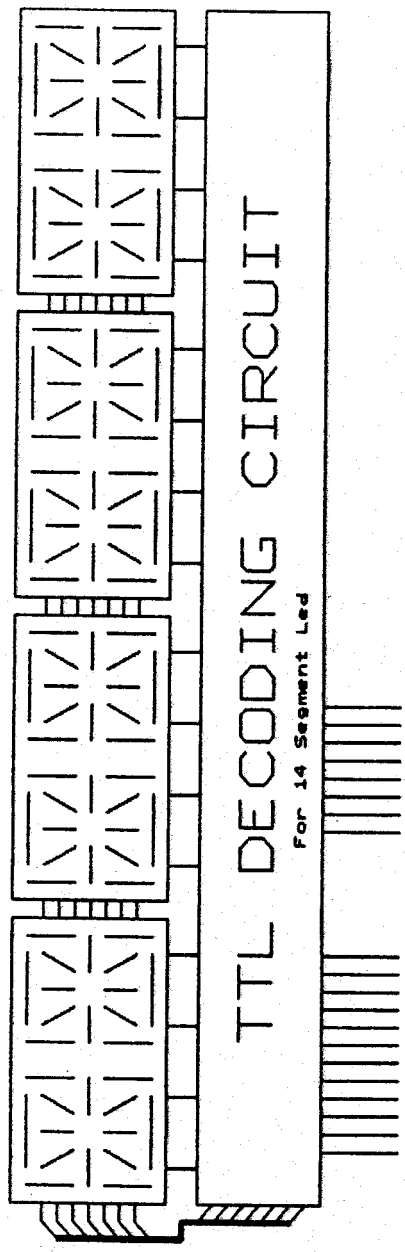


RELEASE VALVE V1
AIR RELEASE RESISTOR

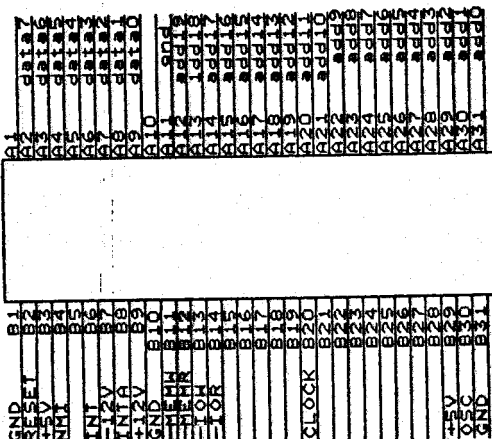
PNEUMATIC SYSTEM		
Size	Document Number	REV
A		PNE
Date:	January 1, 1980	Sheet of



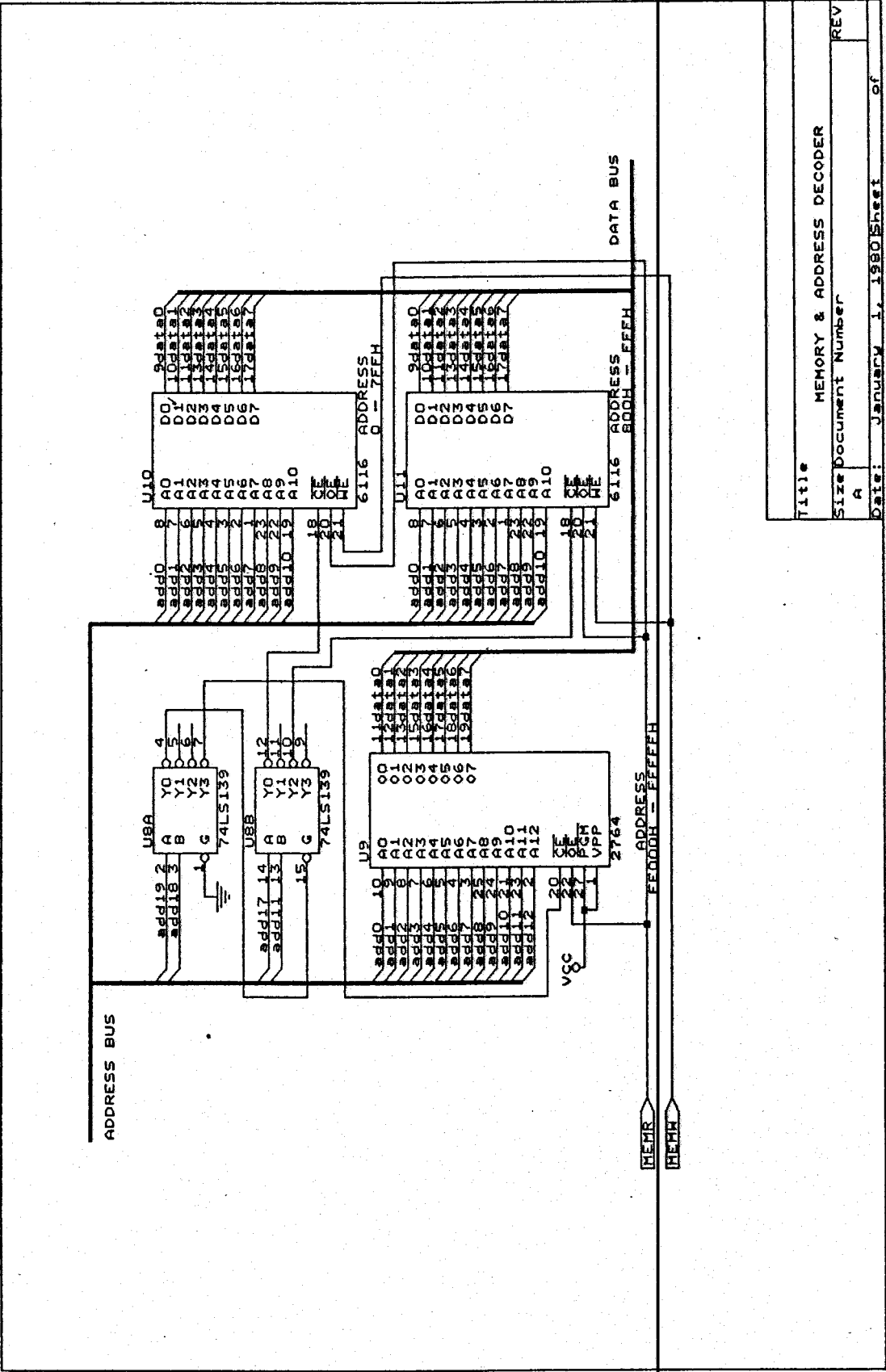
Title		NUMERIK DISPLAY	
Size Document Number		REV	
A			
Date:		January 1, 1980 Sheet	
		of	



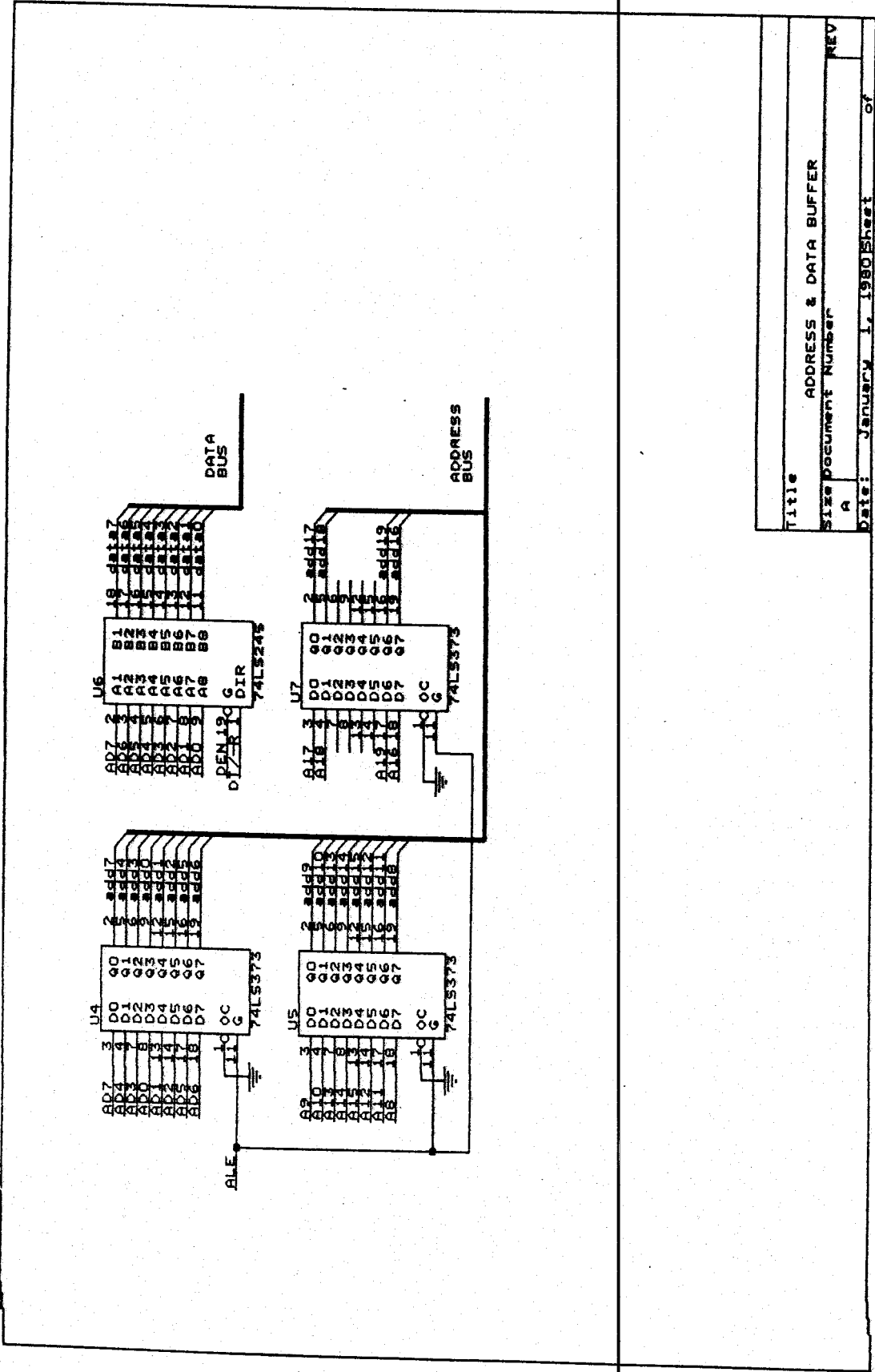
S L O T



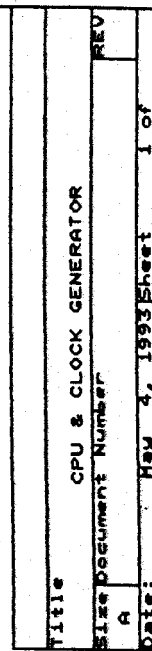
Size Document Number		REV
A		
Date: January 1, 1980		Sheet of

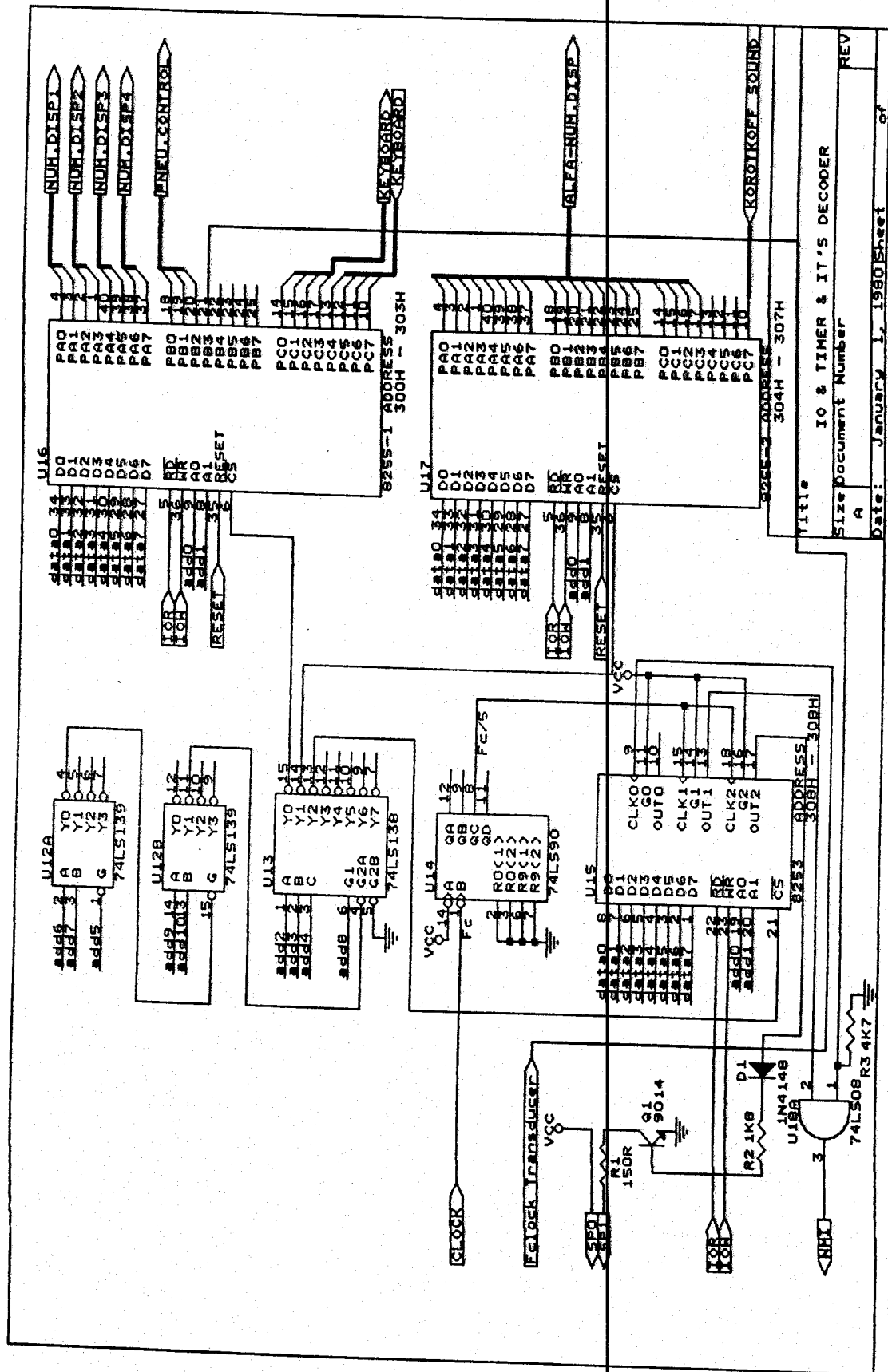


Title		MEMORY & ADDRESS DECODER
Size Document Number		REV
A		
Date: January 1, 1980		Sheet of

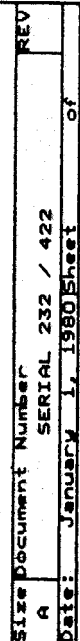


Title		ADDRESS & DATA BUFFER
Size document Number		A
REV		1
Date: January 1, 1980		Sheet of





Title
IO & TIMER & IT'S DECODER
Size Document Number
A
Date: January 1, 1980 Sheet of



INSTITUT TEKNOLOGI SEPULUH NOPEMBER
KAMPUS ITS KEPUTIH SUKOLILO

USULAN TUGAS AKHIR

NAMA / NRP : Agus Susilo / 2882201025
BIDANG STUDI : Elektronika
TUGAS DIBERIKAN : 15 Agustus 1992
TUGAS DISELESAIKAN :
DOSEN PEMBIMBING : 1. Ir. Karyadi, MSc

JUDUL

ANALISA PENGUKUR TEKANAN DARAH PERIODIK, KONTINU, DAN OTOMATIS
YANG DIKONTROL OLEH MIKROKOMPUTER BERBASIS CPU 8088

URAIAN TUGAS AKHIR

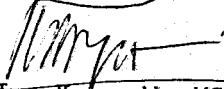
Alat yang direncanakan dimaksudkan untuk mengukur / memonitor tekanan darah manusia secara periodik, kontinu, otomatis. Pengukuran meliputi tekanan darah sistolic dan diastolic. Kecuali itu juga akan diproses untuk menghitung mean artery pressure (MAP), dan heart rate. Alat yang dirancang ini dibandingkan dengan alat sejenis konvensional yang sudah ada memiliki beberapa kelebihan, antara lain karena sifatnya yang periodik, kontinu, dan otomatis, dapat digunakan sebagai sarana pelaksanaan operasi - operasi besar, maupun juga perawatan pasien di ICU.

Di samping itu alat yang direncanakan ini didesain mampu menampilkan hasil diagnosa berdasarkan data - data pengukuran dan data - data awal pasien.

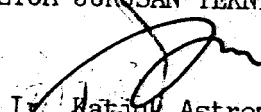
Diharapkan alat yang direncanakan ini memiliki kehandalan yang tinggi karena dikontrol oleh komputer.

MENYETUJUI

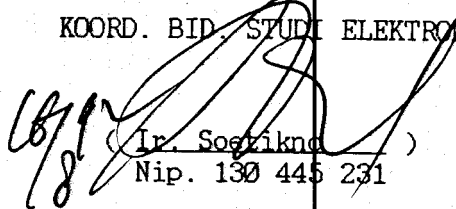
PEMBIMBING


(Ir. Karyadi, MSc)
Nip. 130 520 754

KETUA JURUSAN TEKNIK ELEKTRO ITS


(Ir. Karyadi Astrowulan, MSEE)
Nip. 130 687 438

KOORD. BID. STUDI ELEKTRONIKA


(Ir. Sogrikno)
Nip. 130 445 231

USULAN TUGAS AKHIR

A. JUDUL

ANALISA PENGUKUR TEKANAN DARAH PERIODIK, KONTINU, DAN OTOMATIS
YANG DIKONTROL OLEH MIKROKOMPUTER BERBASIS CPU 8088

B. RUANG LINGKUP

ELEKTRONIKA

RANGKAIAN LINEAR AKTIF

RANGKAIAN TAK LINEAR AKTIF

MIKROELEKTRONIKA

BAHASA ASSEMBLY

ELEKTRONIKA MEDIKA

C. LATAR BELAKANG MASALAH

Penggunaan sistem mikrokomputer untuk mengontrol suatu sistem peralatan mempunyai beberapa keuntungan, seperti tak dibutuhkannya rangkaian digital yang rumit, kerja peralatan dapat dimodifikasi hanya dengan mengubah program / software yang disimpan dalam EPROM. Suatu sistem peralatan yang dikontrol oleh sistem mikrokomputer (minimum sistem) walaupun pengembangan aplikasinya tak seluas dengan peralatan yang diinterfacekan ke komputer IBM PC, namun berkat kesederhanaan rangkaiannya maka minimum sistem lebih ekonomis dan kompak untuk maksud tertentu, pengembangan lebih jauh dari peralatan yang dikontrol oleh minimum sistem dapat dilakukan, misalnya dengan menambah terminal / konektor untuk

dihubungkan dengan komputer IBM PC. Dalam Tugas Akhir ini akan direncanakan dan dibuat peralatan pengukur tekanan darah periodik yang dikontrol oleh mikrokomputer (minimum sistem) dengan intel 8088 sebagai mikroprosesornya, sistem pengukuran melalui sensor tekanan untuk menentukan harga systolic, diastolic, mean artery pressure (MAP), dan denyut jantung per menit (Heart Rate). Peralatan ini juga dirancang dengan tampilan hasil diagnosa tekanan darah seseorang berdasarkan data - data hasil pengukuran, serta data dari pasien seperti usia dan jenis kelaminnya.

D. TUJUAN

1. Memanfaatkan sistem mikrokomputer yang dirancang khusus untuk mengontrol peralatan dengan tujuan tertentu (dalam hal ini pengukur tekanan darah periodik) sehingga terbentuk suatu sistem yang kompak.
2. Memanfaatkan transducer untuk mengukur tekanan.
3. Merancang dan membuat sistem peralatan yang lebih memudahkan pemakainya, dalam hal ini peralatan menampilkan hasil diagnosa berdasarkan pengukuran dan data-data orang yang diukur tekanan darahnya.

E. PENELAAHAN STUDI

1. Mempelajari alat pengukur tekanan darah dengan metoda tak langsung .
2. Mempelajari karakteristik transducer tekanan yang dipakai.
3. Mempelajari sistem pneumatic peralatan pengukur tekanan darah.
4. Mempelajari sistem analog dari peralatan pengukur tekanan darah.
5. Mempelajari sistem mikrokomputer (minimum sistem) yang

akan digunakan untuk mengontrol peralatan pengukur tekanan darah.

6. Mempelajari perangkat lunak (software) yang akan digunakan untuk menjalankan sistem pengukur tekanan darah tersebut.

F. LANGKAH - LANGKAH

1. Studi literatur.
2. Mempelajari cara kerja alat pengukur tekanan darah.
3. Menguji karakteristik transducer tekanan yang akan digunakan.
4. Merancang dan merealisasi sistem pneumatic pengukur tekanan darah.
5. Merancang dan merealisasi bagian analog dan mikrokomputer dari pengukur tekanan darah.
6. Merancang dan merealisasi software yang akan digunakan untuk menjalankan peralatan pengukur tekanan darah tersebut.
7. Mengadakan kalibrasi dan pengukuran pada peralatan yang telah dibuat.
8. Menyusun naskah Tugas Akhir.

G. RELEVANSI

Peralatan pengukur tekanan darah yang direncanakan diharapkan dapat digunakan secara luas, baik oleh tenaga medis yang terlatih (misalnya diruang Intensive Care Unit (ICU), ruang operasi, dan dibagian lain dari klinik atau rumah sakit), maupun digunakan orang yang awam terhadap masalah kesehatan (dalam hal ini tampilan hasil diagnosa membantunya untuk mengetahui kondisi kesehatan orang yang dimonitor tekanan darahnya).

H. JADWAL KEGIATAN

Tugas Akhir ini diharapkan dapat terselesaikan dalam tempo enam bulan dengan perincian kegiatan sebagai berikut:

No.	K e g i a t a n	B u l a n k e					
		1	2	3	4	5	6
1.	Studi literatur	■	■				
2.	Perancangan alat		■	■			
3.	Pengumpulan komponen		■	■			
4.	Pembuatan Hardware & Software			■	■	■	
5.	Kalibrasi dan pengukuran					■	■
6.	Penulisan naskah					■	■

DAFTAR RIWAYAT HIDUP



Agus Susilo dilahirkan pada tanggal 18 Agustus 1968 di Cirebon, anak sulung dari pasangan Susilo dan Yeni Susilo.

RIWAYAT PENDIDIKAN

1. TK Pertiwi Losari. lulus th 1974
2. SDN III Losari-Brebes, lulus th 1981
3. SMPN Tanjung-Brebes, lulus th 1984
4. SMAN I Kodya Yogyakarta, lulus th 1987
5. Masuk Teknik Elektro, FTI, ITS, Th 1988, dan diharapkan lulus pada ujian Tugas Akhir Periode Wisuda Nopember 1993 ini.

Selama belajar di ITS, penulis pernah menjadi asisten praktikum Rangkaian Listrik, dan Elektronika Lanjut II. Di tingkat Institut penulis masuk team divisi Elektrik Mobil Listrik Widya Wahana II th 1991-1993.